NAVAL POSTGRADUATE SCHOOL
Monterey, California

THESIS

COMMUNICATION MODELS IN MOBILE COMPUTING SYSTEMS AND MOBILE AGENTS

by

Refik Tufekcioglu

March 2000

Thesis Advisor:	James Bret Michael
Second Reader:	Bert Lundy

Approved for public release; distribution is unlimited.

REPORT DOCUMENTATION PAGE

Form Approved
OMB No. 0704-0188

Public reporting burden for this collection of information is estimated to average 1 hour per response, including the time for reviewing instruction, searching existing data sources, gathering and maintaining the data needed, and completing and reviewing the collection of information. Send comments regarding this burden estimate or any other aspect of this collection of information, including suggestions for reducing this burden, to Washington headquarters Services, Directorate for Information Operations and Reports, 1215 Jefferson Davis Highway, Suite 1204, Arlington, VA 22202-4302, and to the Office of Management and Budget, Paperwork Reduction Project (0704-0188) Washington DC 20503.

1. AGENCY USE ONLY *(Leave blank)*	**2. REPORT DATE** March 2000	**3. REPORT TYPE AND DATES COVERED** Master's Thesis

4. TITLE AND SUBTITLE: COMMUNICATION MODELS IN MOBILE COMPUTING SYSTEMS AND MOBILE AGENTS

5. FUNDING NUMBERS

6. AUTHOR Tufekcioglu, Refik

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)
Naval Postgraduate School
Monterey, CA 93943-5000

8. PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING / MONITORING AGENCY NAME(S) AND ADDRESS(ES)

10. SPONSORING / MONITORING AGENCY REPORT NUMBER

11. SUPPLEMENTARY NOTES
The views expressed in this thesis are those of the authors and do not reflect the official policy or position of the Department of Defense or the U.S. Government.

12a. DISTRIBUTION / AVAILABILITY STATEMENT
Approved for public release; distribution is unlimited.

12b. DISTRIBUTION CODE

13. ABSTRACT *(maximum 200 words)*

This thesis study covers wired and wireless mobile computing environments, introduces the components of the mobile environment, discusses the constraints of mobility, and contains a taxonomy of the current techniques/models that reduce the overheads associated with wireless mobile communication. One of the goals of this thesis study was to identify and define communication techniques and models that are used by mobile computing systems to minimize wireless communication cost. The following communication techniques and models have been covered in this study: caching, screen caching, differencing, protocol reduction, header reduction, data access profile, delayed writes, strict and loose reads, semantic callbacks and validators, data allocation, data compression, data scheduling, proxy process, adaptation strategy, resource revocation, auto connect/disconnect, and adaptive antennas. The trade-offs between these techniques/models have also been presented. Other goals of this study were to introduce scripts and mobile agents, and explore their security features in mobile computing environments. The usage of mobile agents in military applications has been investigated. Finally, conclusions and recommendations have been provided for wireless mobile computing and mobile agent technology.

14. SUBJECT TERMS Mobile computing, portable computers, mobile environments, mobile agents, intelligent agents, wireless networks, caching, proxy process, adaptation, mobile communication, wireless communication, disconnected operation, energy consumption, cellular communication.

15. NUMBER OF PAGES
156

16. PRICE CODE

17. SECURITY CLASSIFICATION OF REPORT Unclassified	**18. SECURITY CLASSIFICATION OF THIS PAGE** Unclassified	**19. SECURITY CLASSIFICATION OF ABSTRACT** Unclassified	**20. LIMITATION OF ABSTRACT** UL

THIS PAGE INTENTIONALLY LEFT BLANK

Approved for public release; distribution is unlimited

COMMUNICATION MODELS IN MOBILE COMPUTING SYSTEMS AND MOBILE AGENTS

Refik Tufekcioglu
Lieutenant Junior Grade, Turkish Navy
B.S., Turkish Naval Academy, 1994

Submitted in partial fulfillment of the
requirements for the degree of

MASTER OF SCIENCE IN COMPUTER SCIENCE

from the

NAVAL POSTGRADUATE SCHOOL
March 2000

Author: ______________________
Refik Tufekcioglu

Approved by: James B. Michael
James B. Michael, Thesis Advisor

____
Bert Lundy, Second Reader

Dan C Boger
Dan Boger, Chairman
Department of Computer Science

THIS PAGE INTENTIONALLY LEFT BLANK

ABSTRACT

This thesis study covers wired and wireless mobile computing environments, introduces the components of the mobile environment, discusses the constraints of mobility, and contains a taxonomy of the current techniques/models that reduce the overheads associated with wireless mobile communication. One of the goals of this thesis study was to identify and define communication techniques and models that are used by mobile computing systems to minimize wireless communication cost. The following communication techniques and models have been covered in this study: caching, screen caching, differencing, protocol reduction, header reduction, data access profile, delayed writes, strict and loose reads, semantic callbacks and validators, data allocation, data compression, data scheduling, proxy process, adaptation strategy, resource revocation, auto connect/disconnect, and adaptive antennas. The trade-offs between these techniques/models have also been presented. Other goals of this study were to introduce scripts and mobile agents, and explore their security features in mobile computing environments. The usage of mobile agents in military applications has been investigated. Finally, conclusions and recommendations have been provided for wireless mobile computing and mobile agent technology.

TABLE OF CONTENTS

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF FIGURES

THIS PAGE INTENTIONALLY LEFT BLANK

LIST OF TABLES

THIS PAGE INTENTIONALLY LEFT BLANK

ACKNOWLEDGEMENT

It has been my great privilege to work with Prof. James B. Michael, my thesis advisor. I am very grateful for the quality of guidance, support, and enthusiasm that I have received from Prof. James B. Michael during my efforts to conduct this thesis research. I would like to thank Prof. James B. Michael for always helping me and supporting me during the entire thesis process.

I appreciate the support provided by Prof. Bert Lundy, my second reader.

Finally, I would like to dedicate my thesis study to my mother Sukriye Tufekcioglu, and to my father Unal Tufekcioglu, who both lost their lives at the devastating earthquake in Turkey on 17 August 1999 during my thesis study.

THIS PAGE INTENTIONALLY LEFT BLANK

I. INTRODUCTION

A. PROBLEM STATEMENT

Mobile computing is a relatively young area of research and is increasingly becoming commonplace. There has been relatively little work done in mobile computing area compared to the other areas of computer science. Mobile computing involves the movement of physical computing devices such as laptops, palmtops, and wearable computers. There are three aspects of physical mobility: wireless connectivity, weak connectivity, and weak energy autonomy of the mobile computing devices.

Power is one of the most important commodities in mobile communications. Mobile elements rely on a limited energy source. Mobile computers can only operate as long as their batteries maintain power. While battery technology improves over time, the concern for power consumption will not diminish.

Mobile connectivity is highly variable in terms of performance and reliability. Communication bandwidth is likely to remain a major performance bottleneck in the future. A mobile user may have to rely on low-bandwidth wireless connections.

Wireless communication and its cost are a major concern in mobile computing. Wireless communication is much more expensive than wired communication because of the limited bandwidth. The transmission of data over wireless links is slow, unreliable, and expensive. Therefore, reducing wireless communication cost is a very important issue in mobile computing environments.

One of the goals of this thesis study is to identify and define wireless communication models of mobile computing systems in order to minimize wireless communication costs.

A script is a record that consists of a sequence of commands in a text file. The main reason for using scripts in mobile computing is to delegate the mobile user's tasks from a portable computer to a network resource. The mobile user's tasks are designed as scripts by the portable computer and then these tasks sent to a mobile agent in order to enhance the weak flow of mobile communication.

A mobile agent is a component of a software program associated with a mobile user. A mobile agent acts as the mobile user's representative connected to an external server and able to receive requested data even if the mobile user's computer is disconnected from the external network. When the mobile user is re-connected to the external network, the mobile agent returns back to the mobile user's computer. Therefore, mobile agents can save on wireless communication costs and reduce the portable computer's power consumption.

Other goals of this study are to introduce scripts and mobile agents, and to explore their security features in mobile computing environments.

B. OBJECTIVE

The purpose of this thesis study is to identify and define wireless communication models of mobile computing systems in order to minimize wireless communication costs, introduce scripts and mobile agents, and explore the security features of scripts and mobile agents for mobile computing environments.

C. BENEFIT OF STUDY

This research provides conclusions and recommendations for the development of current mobile computing systems.

The resulting recommendations and conclusions also support the Turkish and U.S. Navies Research Centers.

D. SCOPE AND LIMITATIONS

Mobile computing is a broad topic. The focus of this thesis is on wireless mobile computing. Wireless communication techniques for minimizing wireless communication cost, scripts, mobile agents, and mobile agent security issues will be analyzed.

E. OVERVIEW

This thesis is organized in six chapters: (I) Introduction, (II) Background, (III) Communication Models in Mobile Computing systems, (IV) Mobile Code, Scripts, Mobile Agents, and their Security Features, (V) Using Mobile Agents in military applications, (VI) Conclusions and Recommendations.

Chapter II is comprised of two parts. Part A presents wired and wireless mobile computing environments, and introduces the components of the mobile environment and mobility constraints. Part A also contains a taxonomy of the current techniques for minimizing mobile computing communication cost. Part B presents mobile agents and their benefits.

Chapter III introduces communication models in mobile computing systems. Techniques presented for minimizing wireless connection costs are: caching, screen caching, differencing, protocol reduction, header reduction, data access profile, delayed writes, strict and loose reads, semantic callbacks and validators, data allocation, data compression, data scheduling, proxy process, adaptation strategy, resource revocation, auto connect/disconnect, and smart antennas. This chapter also explains the trade-offs between these models.

Chapter IV is comprised of three main parts. Part A presents mobile code and its security features, and explains the benefits of mobile code, programming languages for mobile code, and mobile code security. Part A also introduces the Firewalling, Sandbox, Code Signing, and Proof-Carrying Code approaches for providing assurances against hostile mobile codes. Part B presents scripts and script security, and introduces scripting languages, and the advantages of using remote importable scripts. Part B also explains the use of scripts in mobile computing and the use of cache memory with scripts. Part C presents mobile agents and their security. Part C also discusses mobile agent concepts, the architecture of a mobile agent system, languages for mobile agents, and mobile agent systems.

Chapter V presents the use of mobile agents in military applications. Chapter VI provides conclusions and recommendations.

II. BACKGROUND

A. MOBILE COMPUTING

The goal of mobile computing is to provide mobile users with access to applications and basic communication services in a mobile computing environment. Mobile computing gives mobile users the opportunity to work with other computers from almost anywhere. A mobile computing device can be connected to a wired network with or without wires. Wired connections are more common among the general population and use modems. Wireless connections use radio links to receive and send information. Wireless networking enhances the utility of carrying a computing device. Laptops, palmtops, personal digital assistants (PDA), and other portable computers that easily connect to the Internet and commercial databases are becoming increasingly popular.

1. Mobile Computing Environment

La Porta, T.F., Sabnani, K.K., and Gitlin, R.D. define mobile computing environment in [Ref. 34] shown in Figure 2.1.

The mobile computing environment includes both wired and wireless network connectivity. Mobile users can operate on the various environments such as an office, home, hotel, airplane, or automobile. Mobile users may communicate through a wired network connection or via wireless access, and must contend with variable bandwidths, different link characteristics, and end-devices with varying displays and processing power.

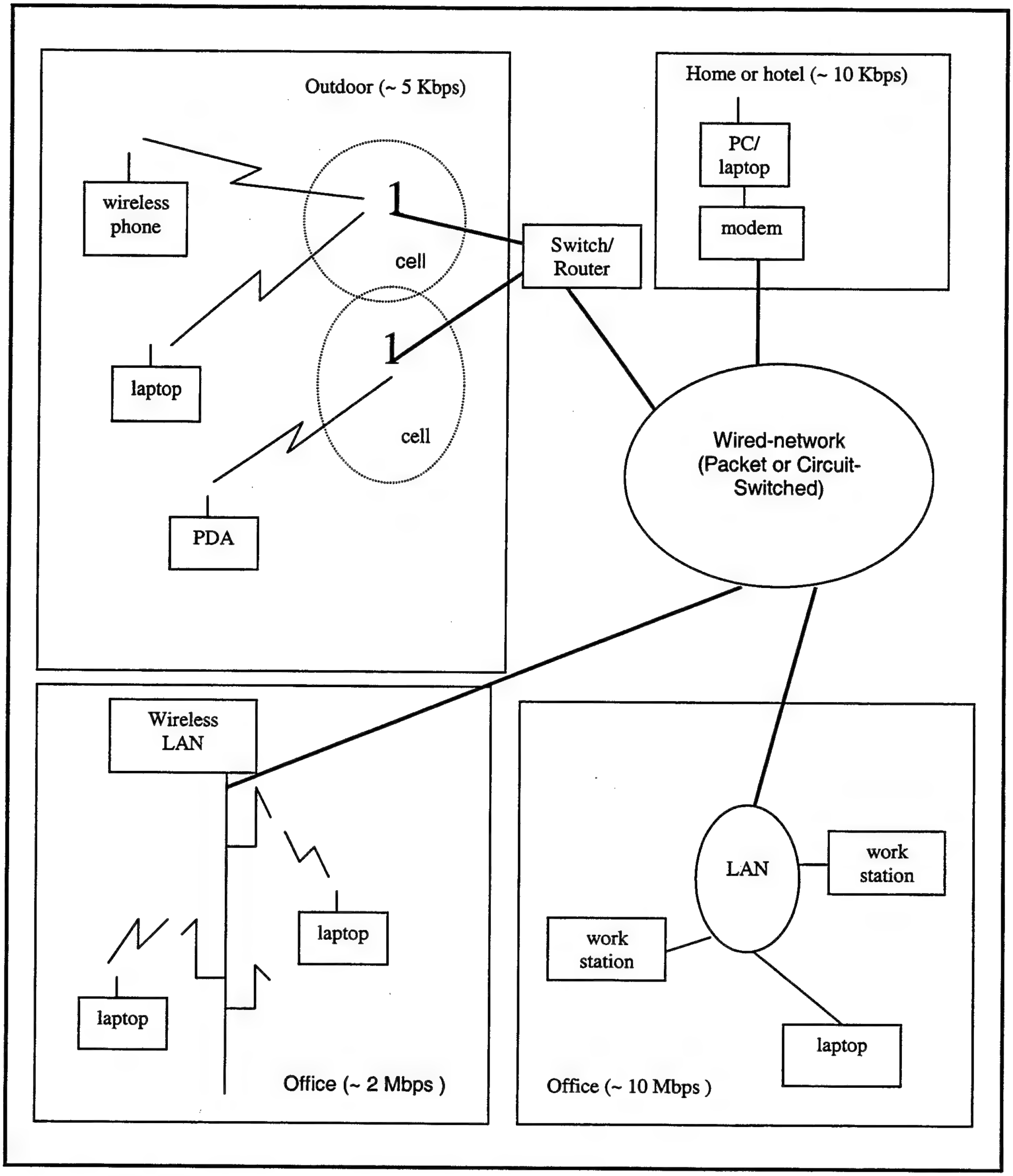

Figure 2.1: Mobile Computing Environment. [From Ref. 34]

a. Wired Mobile Computing Environment

In a wired office environment, network connectivity is achieved through a Local Area Network (LAN). Local communication is inexpensive, and mobile users have access to powerful, high-quality local file servers over wired networks. Local file servers are powerful computers that support high-performance applications. Local Area Networks provide highly available services and high-bandwidth communications.

In a home environment, end-devices are high-powered personal computers (PC). Network connections are made via telephone lines using modems. Telephone connection is expensive, and therefore, users are limited primarily by expense as opposed to decreased computing capabilities.

A mobile user in a hotel room uses a laptop, palmtop, or PDA as an end device and a telephone line for network connection. In a hotel or home environment, end-devices are used while disconnected from the network. In a hotel environment, storage space and local processing power are less abundant than in an office or home environment.

b. Wireless Mobile Computing Environment

Mobile users that may operate in an indoor or in a wide area outdoor environment are connected to a wired network by wireless links. The bandwidth of the wireless links can be much less than that of the wired networks.

Within a building, the connectivity can be through a wireless LAN. In this environment, end-devices will be small with limited memory and power.

Therefore, protocols and applications must be designed to operate at a low bandwidth.

In a wide area outdoor environment, a mobile user can communicate through wireless communication networks. Data rates in wireless networks are low. Wireless services are expensive and have high error rates and frequent disconnections. Therefore, the amount of communication and the type of information exchanged with the end devices must be limited. Mobile users should be able to operate in a disconnected mode and to reconnect to the network periodically due to the limiting factors of wide-area wireless connectivity.

c. Architecture of a Mobile Computing Environment

Mirghafori, N., and Fontaine, A. describe the major elements of a mobile environment in [Ref. 3], shown in Figure 2.2. The major elements of a mobile computing environment are the following:

- **Mobile Host**: A mobile host is a mobile computing device with a cache, disk, and small display screen (i.e., a laptop or a palmtop). The mobile host is capable of wireless communication and is battery operated.

- **Home Server**: A home server is the server on which the mobile host is originally registered, and serves as the permanent storage mechanism of the mobile host files. The home server may physically own a mobile host's pages or just have the capability to retrieve the pages from other servers.

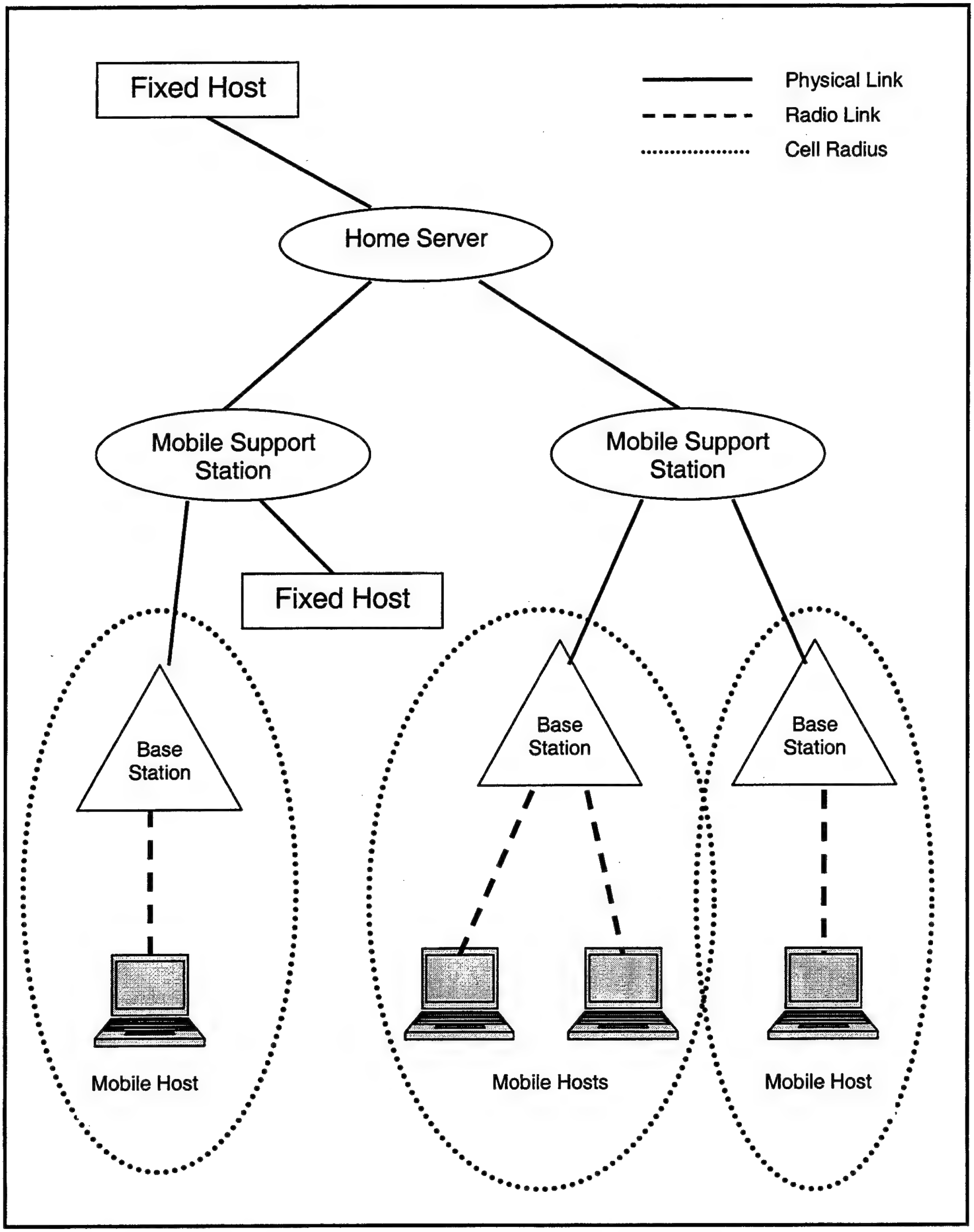

Figure 2.2: Architecture of a Mobile Computing Environment. [From Ref. 3]

- **Mobile Support Station**: A mobile support station is a server that provides services such as cache, RPC requests and retries, etc., to the mobile host. The mobile support station communicates with the mobile hosts within its cell via radio waves. A mobile host registers with a mobile support station upon entering the broadcast range of the mobile support station.

- **Base Station**: A base station broadcasts data messages and does not play a direct role in file access. The base station is conceptually merged with the mobile support station. [Ref. 3]

2. Constraints of Mobile Computing

The purpose of mobile computing is to provide a mobile user with the capacity to communicate with networks from all over the world. Mobile computing must contend with the constraints of both portable computer hardware and wireless communication. The weak flow of information over wireless links and weak energy autonomy due to limited battery power are the major constraints involved in mobile computing.

The key concerns with mobile computing are low bandwidth on wireless networks and network/application performance. Therefore, mobile computing systems and applications attempt to limit the aggregate bandwidth used on wireless links. Network and application performance include reasonable throughput, response time, latency in the presence of low-bandwidth, wireless links, and fast connection establishment in the presence of mobility.

The mobile computing environment differs from a fixed computing environment in many ways. First, mobile users access networks over wireless links. Therefore, mobile users will have access to lower bandwidth and experience higher error rates than wired network users. Second, the wireless links are unreliable in terms of availability. Mobile users may often not be within the coverage area of a network. Therefore, networks and applications must support mobile users that intermittently disconnect from a network. Third, mobile users will not be stationary whether accessing the network by either wired or wireless means. Finally, mobile computing devices will likely have limited processing and power capabilities as compared to desktop computers. [Ref. 34]

Satyanarayanan, M. describes the constraints of mobility in [Ref. 2]. According to Satyanarayanan, mobile computing is characterized by four constraints:

- **Mobile computing elements are resource-poor relative to the static elements.** For a given cost and level of technology, considerations of weight, power, size, and ergonomics negatively impact computational resources such as processor speed, memory size, and disk capacity. While mobile computing elements, such as processor, memory size, and disc capacity, will improve, they may always be inferior to static elements.

- **Mobility is inherently hazardous.** In addition to security concerns, mobile computers are more vulnerable to loss, theft, or damage.

- **Mobile connectivity is highly variable in terms of performance and reliability.** Outdoors, a mobile client may have to rely on a low-bandwidth wireless network with gaps in coverage.

- **Mobile computing elements rely on a finite energy source.** While battery technology will likely improve over time, concern for power consumption will not diminish. In order for power consumption to be fully effective sensitivity to its use must span many levels of hardware and software.

These constraints are not artifacts of current technology, but are intrinsic to mobility. Together, they complicate the design of mobile information systems and require us to rethink traditional approaches to information access. [Ref. 2]

3. Mobile Computing Communication and Its Cost

Wire-based network hardware and software can be expensive. Cost depends on the number of computers in the network and the features of the network. The cost of a wired network, which consists of ten computers and a printer, may range from $2,500 to $35,000. Wired-based networks bring some complexities, such as training network users, training administrators, and adding new software or hardware to the network, to support changes in network requirements. These complexities result in additional cost. In addition, a single failure may cause the entire network to breakdown. It is possible to design networks to be fault tolerant so that they are resistant to breakdowns. Fault tolerance requires having duplicates of critical hardware. Therefore, fault tolerance can also increase the cost of building and maintaining a wired-based network.

The annual cost of owning and running a mobile computing device for business use, including planning, installation, support, and disposal, can run up to $20,000. A company keeps its mobile computing devices an average of 18 months to three years, so the cost of maintaining a mobile computing device actually dwarfs the initial price tag. The high cost of equipping and supporting a mobile device or support staff can erode the advantages conferred by their mobility.

Mobile users have online access to a large number of databases via wireless links. For example, mobile users can access airline schedules, prices of financial instruments, traffic and weather information, etc. The potential market for wireless access is billions of dollars annually in access and communication charges.

Wireless communication is more expensive than wired communication because of the limited bandwidth. For example, cellular telephone call costs about 35 cents per minute. As another example, RAM Mobile Data Corporation charges 8 cents per data message. Wireless communication can become very expensive for mobile users who perform hundreds of transactions per day. Consequently, it is important to minimize wireless communication cost.

Mirghafori, N. and Fontaine, A. explain why mobile computing communication and its costs are a major concern in mobile computing in [Ref. 3]. The authors identify the following issues:

- Wireless communication links are slow and unreliable: on a long-haul radio link, the transmission rate is 19.2 Kbps; radio links tend to have more background noise than “hardwire” links.

- For a mobile host, it is more expensive to send data than to receive data for the following reasons:

  (1) As with cellular telephone service, one pays more service fees to send data than to just receive data,

(2) For a mobile host, sending a message consumes more power than receiving a message. In other words, the communication cost between the mobile support station and the mobile host is asymmetric.

- The dollar cost of sending information over a wireless network is high.

- The overhead of connection setup and teardown is high. Therefore, fewer longer conversations are better than many short conversations.

- There is a high contention on the low-bandwidth wireless link when many mobile hosts try to talk to a mobile support station.

One of the most important commodities in mobile computing communications is power. A mobile computing device can operate as long as its battery maintains power. The trend in mobile computing is moving towards more communication dependent activities with mobile users switching from traditional wired Ethernet communication to wireless communication.

Today mobile computing devices are as powerful as desktop personal computers. This improved technology has been made possible by advances in memory, processor, and integration technologies. However, battery technology has not made similar improvements. Energy density of batteries has only doubled in the last 35 years whereas processor speed keeps doubling every 18 months. The slow improvement in battery lifetime shows that energy consumption will be one of the most important factors in designing systems and wireless networking support for mobile computing devices in the future.

A new type of battery powered by a cathode made of an unusual form of iron absorbing more electrons, boosts the battery's power and provides 50 percent more power. The new batteries rely on "super iron" because of their enhanced ability to store a charge and provide more power. Researchers have also developed a rechargeable version, which uses the super-iron cathode and an anode of the same material as in other rechargeable batteries [Ref.38].

Some software and hardware techniques have been proposed to reduce the power consumption of a mobile computing device during operation. Software-level solutions focus on modulating the power used by the mobile transmitter during active communication, whereas hardware-level techniques concentrate on non-communication components such as processors, disks, and displays. The goal of these techniques is to predict when the mobile computing device will not be used and to suspend it for those periods.

4. Taxonomy of the Current Techniques/Models for Minimizing Mobile Computing Communication Cost

Table 2.1 presents the current techniques/models for minimizing mobile computing communication cost. The following presents a summary of the papers, which have been published on the costs of mobile computing communication:

REF. MODEL	1	2	3	4	5	6	7	8	9	10	15	17	34	39	40	41	42
Caching	√	√	√	√	√	√	√	√	√			√	√		√		
Screen Caching	√																
Differencing	√																√
Protocol Reduction	√																√
Header Reduction	√																√
Data-Access Profile			√														
Delayed Writes	√		√												√		
Strict-Loose Reads			√														
Semantic Callbacks and validators		√															
Data Allocation				√				√									
Data Compression	√															√	
Data Scheduling										√							
Proxy Process			√			√						√	√				
Adaptation Strategy		√												√			
Resource Revocation		√															
Auto connect/disconnect	√				√				√								
Smart Antennas											√						

Table 2.1: Taxonomy of the current techniques / models, which have been used in mobile computing environments for minimizing mobile communication cost.

a. Larry Francis, "Mobile computing: a fact in your future," paper presented at the 15th annual international conference on Computer documentation, pp. 63-67, 1997. [Ref. 1]:

Mobility means reduced connection speed and increased connection cost. Technology to the rescue from high communication cost is cheaper and faster data.

In this paper, Francis proposes the following techniques for reducing mobile communication cost:

- Data compression,
- TCP header reduction,
- Screen caching,
- Screen differencing,
- Queuing requests,
- Latency reduction,
- Protocol reduction,
- Auto connect / disconnect.

The author describes how mobile computing technology interacts with some of the latest developments in computer technology such as network computing, Java, and palmtops. With network computing, mobile computer has little internal data or intelligence. Mobile users might use a network computer built into their airplane seat or hotel room. Network computing needs a cheap, fast way to download programs and data to users, but mobile use makes downloading awkward and expensive. The caching and efficiency techniques applied to Web pages also support the downloading and re-use of Java applets. Personal digital assistants such as palmtops and wearable computers are similar to laptops but are much smaller. These computers make tradeoffs between size, weight, and function, thus sharing the similar advantages and suffering from the similar shortcomings. Although personal digital assistants serve a different population than laptops, they comprise an important part of the mobile computing environment. Palmtops are unlikely to compute significantly with laptops, but interest in them may increase the mobile use environment.

b. Satyanarayanan, M., "Fundamental Challenges in Mobile Computing," paper presented at the 15th annual ACM symposium on Principles of distributed computing, pp. 1-7, 1996. [Ref. 2]:

Satyanarayanan attempts to answer the question, "*What is unique and conceptually different about mobile computing?*" The paper describes a set of constraints intrinsic to mobile computing, and examines the impact of these constraints on distributed systems design.

Satyanarayanan proposes an adaptation strategy, because according to the author, adaptation is the key to mobility. Adaptation insulates users from the vagaries of mobile environments by using local resources to reduce communication. His research explores two different approaches to adaptation, application-transparent and application-aware.

Finally, the author describes opportunities for future research in mobile computing: caching metrics, semantic callbacks and validators, resource revocation, analysis of adaptation, and global estimation from local observations.

c. Mirghafori, N.; Fontaine, A., "A design for file access in mobile environment," paper appears in Mobile Computing Systems and Applications, pp. 57-62, 8-9 December 1994. [Ref. 3]:

The need to reduce communication cost is needed in the mobile environment because transmission of data over radio links is slow, expensive, and unreliable. Providing data consistency is crucial because many mobile applications are database applications, which rely on consistent data. In this paper, the authors propose a design for a file access mechanism that is specific to a mobile environment. The two main design goals are to reduce communication cost and to provide data consistency. They propose to minimize the amount of wireless communication by extensive use of caching, profile information, a proxy process, delayed writes, and making use of loose-reads. Data consistency is provided by using proxy services, a centralized data manager with callbacks, and strict-reads.

Mighafori and Fontaine concentrate their design in reducing the effects of the narrow bandwidth with the rationale that bandwidth is more likely to remain a major performance bottleneck in the future. Communication between the mobile host and the mobile support station is kept at a minimum to achieve a reasonable response time through extensive use of caching and liberal delegation of tasks to the proxy. Caching combined with loose-read capability can reduce network traffic significantly, since many of the mobile users' reads will be performed locally. The authors conclude that the proxy combined with caching is the most significant means for achieving reduction in communication costs.

d. Prasad Sistla, A.; Wolfson, O.; Yixiu Huang, "Minimization of communication cost through caching in mobile environments," paper appears in Parallel and Distributed Systems, IEEE Transactions on, pp. 378-390, April 1998. [Ref. 4]:

In this paper, the authors present and analyze various static and dynamic data allocation methods in order to minimize mobile computing communication. An allocation method determines whether the allocation scheme changes over time or not. In a static allocation method, the allocation scheme does not change over time, whereas, in a dynamic scheme it does. The authors analyze two static allocation methods using the one-copy scheme and the two-copies scheme as well as a family of dynamic data allocation methods. These methods are suggested by the need to select the allocation scheme according to the read/write ratio. The family consists of all the methods that allocate and de-allocate a copy of a data item to the mobile computer based on a sliding window of k requests. The allocation scheme is dynamically adjusted according to the relative frequencies of reads and writes. The algorithms in this family are distributed, and they are implemented by software residing on both the mobile and stationary computers.

The authors' objective is to optimize the communication cost between a mobile computing device and the stationary computer that stores the online database. Their analysis is performed in two cost models. One is connection (or time) based, as in cellular telephones, where the user is charged per minute of cellular telephone connection. The other is message based, as in packet radio networks, where the user is charged per message. Their analysis addresses both the average case and the worst case for determining the best allocation method.

e. Kravets, R.; Krishan, P., "Power Management Techniques for Mobile Communication," paper presented at the 4th annual ACM/IEEE international conference on Mobile computing and networking, pp. 157-168, 1998. [Ref. 5]:

In the mobile computing environment, power is a limited resource. Therefore, users of mobile communication devices need to be conscious of this limitation and conserve energy. The research presented in this article focuses on software-level techniques for managing the mobile host's communication device through suspension of the device during idle periods in communication. Kravets and Krishnan present a novel transport-level protocol for managing the suspend/resume cycle of the mobile host's communication device in an effort to minimize power consumption.

The authors target on the transport layer, where they provide a set of mechanisms that allow communication to be suspended and resumed. Kravets and Krishnan assume a model where the mobile host is communicating with the rest of the network through a base station. This base station may be a proxy, or it may be the connection point for end-to-end communication with other hosts. Often, dealing with mobility does not fit into the standard seven-layer model. By exposing power management techniques to the application, the authors provide a system-level solution aimed at end-to-end communication. The authors concentrate on the communication between the mobile host and the base station, and for clarity assume that all communication to and from the mobile host is directed through one specific base station.

The protocol achieves power savings by selectively choosing short periods to suspend communications and shut down the device. The protocol also manages the important task of queuing data for future delivery during periods of communication suspension, and decides when to restart communication.

f. Lauzac, S. W.; Chrysanthis, P. K.; Tjoa, A. M.; Wagner, R.R., "Programming views for mobile database clients," this paper appears in Database and Expert Systems Applications, proceedings, 9th International Workshop on, pp. 408-413, 26-28 August 1998. [Ref. 6]:

Within a database mobile environment, cached data on mobile clients can take the form of materialized views. In order to efficiently maintain materialized views while respecting disconnected operations, the authors present a mechanism in the form of a proxy within the fixed network. This proxy can assume different roles in order to provide a customizable client-oriented "data warehouse" mechanism which they call the view holder.

A view holder is not static or generic, and maintains a state with respect to the individual mobile clients it supports. When a view holder is required to maintain a particular view, the view specification can be seen as a program specification. A view holder can be programmed to maintain multiple versions of a view in order to compensate for the data changes that occurred to the materialized views that were used during disconnection and present how the view holder allows for efficient interactions with the data sources as well as the mobile hosts. The authors extend the SQL create view statement and show how it can be used to program the view holders.

This paper addresses the problem of caching and maintaining data within a mobile environment in the form of a materialized view. The authors' main contribution is the development of the view holder, a mechanism that maintains customizable versions of cached views specified by an extension of SQL.

g. Chan B. Y.; Si, A; Leong, H. V., "Cache Management for Mobile Databases: Design and Evaluation," this paper appears in Data Engineering, proceedings., 14th International Conference on, pp. 54-63, 23-27 February 1998. [Ref. 7]:

Communication between mobile users and database servers in a mobile computing environment is via wireless channels with low bandwidth and low reliability. A mobile user could cache its frequently accessed database items into its local storage in order to improve performance of database queries and availability of database items for query processing during disconnection.

In this paper, the authors describe a mobile caching mechanism for a mobile environment utilizing a point-to-point communication paradigm. In particular, they investigate issues of caching granularity, coherence strategy, and replacement policy of mobile caching.

The authors present a framework for caching mechanism as one way to improve data access performance in a mobile environment. The caching mechanism is illustrated and implemented on an object-oriented database model. The authors show that page-based caching is not suitable in a mobile context and propose three different caching granularities, namely, attribute caching, object caching, and hybrid caching. The authors also show that conventional cache coherence and replacement schemes are not effective, and propose modified strategies that adapt to object access patterns.

h. Wolfson, O.; Yixiu Huang, "Competitive analysis of caching in distributed databases," paper appears in Parallel and Distributed Systems IEEE Transactions on, pp. 391-409, April 1998. [Ref. 8]:

Wolfson and Huang introduce a model for evaluating performance data allocation and replication algorithms in distributed databases. The model is comprehensive in the sense that it accounts for input output cost, for communication cost, and because of reliability considerations, for limits on the minimum number of copies of the object.

In modern distributed databases, particularly in mobile environments, processors will dynamically store objects in their local database and will relinquish them. Caching is a particular form of dynamic allocation in which a processor that reads an object saves a copy of that object and, thus, it joins the allocation scheme. The main goal of this paper focus on caching that is discussed in the larger context of dynamic allocation. The authors study caching in a peer-to-peer environment rather than a client server environment.

In this paper, the authors analyze the cost of servicing a set of read-write requests for a replicated object. This set of requests is usually ordered by some concurrency-control mechanism such that each read request accesses the most recent version of the object. The cost of servicing a read or write request depends on the allocation scheme of the object, namely, the set of processors that store the most recent version of the object in their local databases. The authors introduce an algorithm for automatic dynamic allocation of replicas to processors.

The allocation scheme of an object is either dynamic or static, namely, it changes as the read-write requests are executed or it remains fixed. The reason for changing the allocation scheme is that the larger the allocation scheme, the smaller the cost of an average read-request and the bigger the cost of an average write request. Thus, in a read-intensive environment, a large allocation scheme is mandated, whereas, in a write-intensive environment, a small allocation scheme is mandated.

i. Barbara, D.; Imieli, T., "Sleepers and workaholics: caching strategies in mobile environments," ACM SIGMOD Record, v. 23, No. 2, pp. 1-12, June 1994. [Ref. 9]:

Caching of frequently accessed data items is an effective technique that reduces communication on the narrow bandwidth wireless channel. However, cache invalidation strategies are severely affected by the disconnection and mobility of clients. The server may no longer know which clients are currently residing under its cell and which are not.

In this paper, Barbara and Imieli propose a taxonomy and analyses of different cache invalidation methods, and study the impact of disconnection time on performance. The authors show that caching is a widely used technique, which may lead to improvement in overall throughput by making it possible to answer queries locally without competing for the scarce wireless bandwidth.

The authors determine that for the mobile computers, which are often disconnected, the best cache invalidation strategy is based on signatures previously used for efficient file comparison. On the other hand, for the mobile computers that are connected most of the time, the best cache invalidation strategy is based on the periodic broadcast of changed data items.

j. Yon Dohn Chung; Myoung Ho Kim, "QEM: a scheduling method for wireless broadcast data," paper appears in Database Systems for Advanced Applications, Proceedings., 6th International Conference on, pp. 135-142, 19-21 April 1999. [Ref. 10]:

In mobile distributed systems, the data on air can be accessed by a large number of clients. In this paper, the authors describe the way clients access the wireless broadcast data with short latency. They define and analyze the problem of wireless data scheduling.

Chung and Kim investigate a wireless broadcast data scheduling method that finds a broadcast schedule of data for reducing the access time of the queries issued by mobile clients. They give a method named QEM, which constructs the broadcast schedule by expanding each query's data set. The proposed method QEM reduces the access time by efficient scheduling wireless broadcast data. The authors study the performance of the QEM within several environmental parameters. QEM effectively constructs a wireless broadcast schedule resulting in a 20 percent reduction in access time.

k. Flinn, J,; Satyanarayanan, M., "PowerScope: a tool for profiling the energy usage of mobile applications, "paper appears in Mobile Computing Systems and Applications, Proceedings., Second IEEE Workshop on, pp. 2-10, 25-26 February 1999. [Ref. 11]:

Energy is a critical resource for mobile computers. In this article, Flinn and Satyanarayanan describe the design and implementation of PowerScope, a tool for profiling energy usage by applications. PowerScope maps energy consumption to program structure, in much the same way that CPU profilers map processor cycles to specific processes and procedures.

The authors' approach combines hardware instrumentation with the kernel software support to measure the current-level energy usage to perform statistical sampling of system activity. Post-processing software maps the sample data to program structure and produces a profile of energy usage by process and procedure.

Attributing costs in detail enables attention to be focused quickly on problem areas in code. Therefore, the authors' most important design consideration is to enable PowerScope to gather sufficient information to produce a detailed picture of system activity, as the usefulness of a profiling tool is directly related to how definitively it assigns costs to specific application events.

l. Barbara, D., "Mobile computing, and databases-a survey," paper appears in Knowledge and Data Engineering, IEEE Transactions on, pp. 108-117, January-February 1999. [Ref. 12]:

Barbara surveys the impact that mobile computing has had in the area of data management. The author first analyzes each of the distinct features of mobile computing and how they affect the implementation of databases for mobile computers, creating new opportunities for research. These distinct features are asymmetry in the communications, frequent disconnections, power limitations, and screen size. Each one of these features has an impact on how data can be effectively managed in a system with mobile clients.

The communication asymmetry, along with the restriction in power that the mobile units have, make the model of broadcasting data to the clients instead of waiting for the clients to request specific data items, an attractive proposition. This is called data dissemination. The limited bandwidth and the pattern of frequent disconnections have a clear impact on how transaction management is implemented and how data consistency is guaranteed in the mobile environment. Screen and power limitations have an impact on the kind of interfaces that can be implemented for data browsing and querying.

m. Chakrabarti, S.; Dutta, G., "A Low Deviation Digital Modulation Scheme for Mobile Communication," paper appears in Personal Wireless Communication, IEEE International Conference on, pp. 193-197, 17-19 February 1999. [Ref. 13]:

In this paper, the authors propose a modulation scheme that can be applied to severely band-limited channels such as land-mobile and satellite-mobile radio channels.

The authors also describe the modulation and demodulation strategy, discuss the performance of the demodulation technique, and state the possible decision strategies that can provide the best results for the use of bandwidth. Chakrabarti and Dutta conclude that the modulation scheme utilizes the correlation property of the data sequence and produces continuous-phase, bandwidth-efficient waveforms with constant envelope.

n. Goyal, A.; Sundareshan, M. K., "Performance analysis of a person-based mobility management scheme for PCN," paper appears in Performance, Computing and Communications Conference, IEEE International, pp. 97-103, 10-12 February 1999. [Ref. 14]:

Mobility management is an issue of central and unique importance in wireless communication systems. Factors that determine the efficiency of a mobility management scheme include the number of times that locations need to be updated, the amount of overhead data that represents mobility-related information, and storage mechanisms that allow for fast storage and retrieval of this information.

In this paper, the authors outline a mobility management scheme that provides a globally unique personal number. A mobile computer can be viewed as an extension of a fixed system and its location changes with time. A geography-independent personal number provides this extension. They perform an analysis of the scheme by using query and update operations as the metrics for determining average call delay and controlling data storage and transmission requirements. This analysis provides a tool for determining network management requirements in a person-based number scheme. A person-based numbering scheme provides an improved mobility management in personal communication networks.

o. Tsoulos, G. V., "Smart antennas for mobile communication systems: benefits and challenges," paper appears in Electronics & Communication Engineering Journal, pp. 84-94, April 1999. [Ref. 15]:

This paper provides an overview of the potential benefits and challenges of applying smart-antenna technology to mobile communication systems.

The author presents an overview of smart antennas in terms of key characteristics, options, challenges, and benefits, in the context of current, but also with a view towards future generation personal communication systems.

Tsoulos notes that communication systems will exploit different advantages or mixtures of advantages offered by smart antennas depending on the maturity of the underlying system. The author concludes that technology advanced systems will be able to benefit most from smart antenna systems. Communication costs can be reduced by exploiting the range extension capabilities of smart antennas. The author states that costs can be further decreased by avoiding extensive use of small cells and instead exploiting the capability of smart antennas to increase capacity where there is a demand for increased capacity.

p. Bhagwat, P.; Bisdikian, C.; Korpeoglu, I.; Krisha, A.; Naghshineh, M., "System Design Issues for Low-Power, Low-Cost Short Range Wireless Networking," paper appears in Personal Wireless Communication, IEEE International Conference on, pp. 264-268, 17-19 February 1999. [Ref. 16]:

In this paper, the authors present the design of a short-range wireless networking system called BlueSky developed at IBM Research to address the challenges of providing low-cost, low-power, indoor wireless networking to handheld devices. The authors show that the optimization objectives for short-range indoor wireless systems are quite different from those of traditional cellular wireless systems. The authors also argue that in the next millennium the primary optimization criteria for the design of short-range wireless systems will shift from the traditional spectral efficiency towards battery lifetime and cost.

r. Carlier, D.; Trane, P., "Task delegation model assigned to mobile computing," paper appears in Information, Communications and Signal Processing, 1997. ICICS., Proceedings of 1997 International Conference on, v. 1, pp. 220-224, 9-12 September 1997. [Ref. 17]:

This paper aims at describing task role and delegation even if the task is not completely and precisely defined before the sending phase of tasks. One of the main reasons for using scripts in mobile computing is to be able to delegate tasks from a portable terminal to a workstation with more resources. An initial proposal is to assign each user to a representation agent on the wired network. Tasks designed as a script are sent to this agent to take advantage of network opportunities. The missing information on the workstation is imported from distant servers through remote script import protocols. The authors present cache architecture to make this operation as efficient as possible. In addition, the authors also discuss security features according to the main characteristics of the different kinds of scripts, namely personalized, certified, and protected.

In this paper, Carlier and Trane represent a general description and implementation of scripts to prove the feasibility of the mobile computing system. As mobile computers are obviously less efficient than powerful computers located on a wired network and wireless communications are not as easy and cheap, it is important to introduce scripts on the wired network, which can help reducing wireless traffic. Scripts are considered as task executors.

B. MOBILE AGENTS

A mobile agent is a software program that accepts tasks from the mobile user, to help overcome drawbacks associated with the mobile computing environment.

Mobile agents can navigate independently over heterogeneous networks submitted by mobile users to fulfill tasks. Mobile agents are special mobile code[1] entities and are composed of code and explicit data elements. Mobile agents are very useful for mobile computing. A mobile agent is an interface between the mobile user and external networks acting as the mobile user's representative. The mobile agent is always connected to the related external network server and is able to receive requested data even if the mobile user's computer is disconnected. After a mobile agent is submitted, a mobile user can be disconnected from the network. When it reaches its owner computer, the mobile agent is capable of interacting with external network servers, moving to another network server, and resuming execution.

Lange, D.B. and Oshima, M. identify the following main benefits of using mobile agents in [Ref. 18]:

- **Mobile agents reduce the network load.** Mobile agents allow users to package a conversation and dispatch it to a destination host where interactions take place locally. Mobile agents are also useful when reducing the flow of raw data in the network. When very large volumes of data are stored at remote hosts, that data can be processed in its locality rather than transferred over the network.

[1] Mobile code denotes the programs that are executed on foreign computers. Mobile codes can be written by anyone and execute on any machine that runs a browser. Examples of mobile codes are Java Applets and Postscript.

- **Mobile agents overcome network latency.** Critical real-time systems, such as robots in manufacturing processes, need to respond in real time to changes in their environments. Controlling such systems through a factory network of substantial size involves significant latencies, and for critical real-time systems, such latencies are unacceptable. Mobile agents offer a solution, because they can be dispatched from a central controller to act locally and execute the controller's directions without delay.

- **Mobile agents encapsulate protocols.** When data is exchanged in a distributed system, each host owns the code that implements the protocols needed to properly code outgoing data and interpret incoming data. However, as protocols evolve to accommodate new requirements for efficiency or security, it is cumbersome if not impossible to upgrade protocol code properly. As a result, protocols often become a legacy problem. On the other hand, mobile agents can move to remote hosts to establish communication based on protocols.

- **Mobile code executes synchronously and autonomously.** Mobile devices often rely on expensive or fragile network connections. Tasks requiring a continuously open connection between a mobile device and a fixed network are usually neither economically nor technically feasible. To solve this problem, tasks can be embedded into mobile agents, which can then be dispatched into the network. After being dispatched, the agents become independent of the process that created them and can operate asynchronously and autonomously.

- **Mobile agents adapt dynamically.** Mobile agents can sense their execution environment and react autonomously to changes. Multiple mobile agents have the unique ability of distributing themselves among the hosts in the network to maintain the optimal configuration for solving a particular problem.

- **Mobile agents are naturally heterogeneous.** Network computing is fundamentally heterogeneous, often from both hardware and software perspectives. A heterogeneous network includes computers and devices from various manufacturers and transmits data using more than communications protocols. Mobile agents provide optimal conditions for a seamless system integration, because they are generally computer and transport-layer independent.

- **Mobile agents are robust and fault-tolerant.** A mobile agents' ability to react dynamically to unfavorable situations and events makes it easier to build robust and fault-tolerant distributed systems. If a host is being shut down, all agents relying on that machine are warned and given time to dispatch and continue their operation on another host in the network.

- **E-commerce.** Mobile agents are well suited for e-commerce. A commercial transaction may require real-time access to remote resources, such as stock quotes and perhaps even agent-to-agent negotiation. Different agents have different goals and implement and exercise different strategies to accomplish them.

- **Personal Assistance.** Mobile agents' ability to execute on remote hosts makes them suitable assistants to perform tasks in the network on behalf of their creators. Remote assistants operate independently of their limited network connectivity; their creators can even turn off their computers.

- **Secure Brokering.** The parties can let their mobile agents meet on a mutually agreed secure host where collaboration takes place without the risk of the host siding with one the visiting agents.

- **Distributed Information Retrieval.** Instead of moving large amounts of data to the search engine so the search engine can create search indexes, the agent creator can dispatch mobile agents to remote information sources where mobile agents locally create search indexes that can later be shipped back to the system of origin.

- **Telecommunication Networks Services.** The support and management of advanced telecommunication services are characterized by dynamic network reconfiguration and user customization. The physical size of these networks and the strict requirements under which they operate call for mobile agent technology to function as the glue that keeps the systems flexible yet effective.

- **Workflow applications and groupware.** The nature of workflow applications includes support for the flow of information among coworkers. Mobile agents are especially useful here, because in addition to mobility, they provide a degree of autonomy to the workflow item.

- **Monitoring and notification.** An agent can monitor a given information source without being dependent on the system from which it originates. Agents can be dispatched to wait for certain kinds of information to become available.

- **Information dissemination**. Mobile agents embody the so-called Internet push model. Agents can disseminate information, such as news and automatic software updates, for vendors.

- **Parallel processing.** If a computation requires so much processor power that it must be distributed among multiple processors, an infrastructure of mobile agent hosts can be a plausible way to allocate the related processes. [Ref. 18]

THIS PAGE INTENTIONALLY LEFT BLANK

III. COMMUNICATION MODELS

A. COMMUNICATION MODELS IN MOBILE COMPUTING SYSTEMS

1. Caching

In a mobile computing environment, the limited power of the mobile computing device and the low bandwidth of the wireless communication link are the major constraints of wireless mobile operation. Focusing on the low bandwidth constraints leads to higher-end mobile computing devices with caches.

Caching plays a very important role in mobile computing. Caching can alleviate the limitations on the performance and availability of the weakly connected mobile operation. Caching of data in a mobile computing environment limits communication and improves the performance and availability of services such as browsing applications. Effective caching can significantly reduce wireless network traffic. Mobile users can access data more quickly if that data is cached. Caching can also speed data access by filtering out approximately fifty percent of the message traffic.

In the mobile computing environment, one utilizes a point-to-point communication paradigm, referred to as *mobile caching.* The main goal of mobile caching is to achieve a reasonable response time by keeping the communication between the mobile computing device and the mobile support station at a minimum. Caching minimizes wireless communication traffic because mobile users perform read operations locally.

A caching mechanism is characterized by its caching granularity, cache coherence strategy, and cache replacement policy, and each of these are defined as follows:

Caching Granularity. There are three different levels of caching granularity: attribute caching, object caching, and hybrid caching. In attribute caching, frequently accessed attributes of objects are cached in a mobile user's local storage. In object caching, the objects themselves are cached. In hybrid caching, only the frequently accessed attributes of accessed objects are cached.

Cache Coherence. A cache coherence strategy involves cache invalidation and update schemes to invalidate and update an outdated cached object. (Cached objects become outdated when the objects residing at the server are updated.) A cache coherence strategy provides a mobile computing device with the ability to update its cached objects. A mobile user should take an active role in maintaining the coherence of the cached objects and determining if a particular cached item should be invalidated. It is not feasible for the mobile user's home server to keep track of all cached copies of individual items.

Cache Replacement. If a mobile user can provide unlimited disk storage on his mobile computing device, all the frequently accessed objects of the mobile user can be cached. However, storage for caching on mobile computing devices is limited. Moreover, caching rarely accessed objects will result in a waste of system resources. Therefore, for best performance, a cache replacement policy is needed to retain only frequently accessed objects.

2. Screen Caching

Internet Browsers can cache screens and store screens internally for the duration of an Internet session. Browsers on mobile computing devices are often set to cache screens such as menu screens, which are frequently used by the mobile user, in order to minimize the amount of wireless mobile communication.

However, problems can arise because the cached screen may become obsolete at unpredictable times. Caching software is used on mobile computing devices to solve this problem. Caching software takes requests for a specific screen and then checks its local cache for that screen. The software can determine whether the screen in the cache memory of the mobile computing device is the most recent or if the screen needs to be replaced and refreshed by comparing time stamps on the screens.

3. Differencing

The concept of differencing means to cache an object on both the mobile computing device and on its home server. When the home server receives a response from a Web server, the home server computes the difference between the response and the object, and then sends the difference to the mobile computing device. The mobile computing device merges the difference with the original object to create the browser response.

Differencing is also used to distinguish Hyper Text Markup Language (HTML) documents. HTML enables mobile users to enter data and then submit the data for processing by some executable program located on the Web. The executable program on the Web is identified by a Universal Resource Locator

(URL), and then a command is sent from the browser to the server specified by the URL.

The rules for invoking and enabling programs to read data and generate responses are collectively called Common Gateway Interface (CGI). The term CGI Processing refers to the process of executing programs from Web browsers. Caching techniques do not help in CGI processing, because mobile users enter different data for different requests, receive different results, and no two replies to requests are identical for the same URL. A different form of caching and differencing technology is used to minimize responses from CGI programs. This approach is based on the observation that various responses from the same Web server are usually very similar.

Initially, there is no record of a cached response for the URL at the mobile computing device cache memory. The mobile computing device determines that the Hyper Text Transport Protocol (HTTP) request is a CGI request, if the URL is followed by a name/value parameter list. Then, the mobile computing device sends the request to its home server. The home server forwards the mobile user's request to a destination server on the Web. Once the home server receives the response from the destination server, it caches the response before forwarding it to the mobile computing device. Likewise, the object is cached at the mobile computing device before it is sent to the browser. An object has been established for the CGI URL at this point in time.

When the mobile computing device detects a request for CGI processing, the device checks whether the URL is cached. Then, the mobile computing device forwards the user's request to its home server along with the Cyclic Redundancy Check (CRC), value of the object. The CRC is maintained as part of the request state. The HTTP data stream is forwarded to the HTTP server to execute the request.

Once the home server receives a report from the HTTP server, the home server determines that differencing is possible because an object for the URL exists in the cache, and its CRC matches the one received with the request from the mobile computing device. The differencing engine computes the difference stream between the received response and the object. Then, the home server sends the difference stream to the mobile computing device. The mobile computing device's update engine uses the difference stream and constructs a new report for the browser. Consequently, the mobile computing device sends the new report to the browser.

Wireless communication links are slow and unreliable. Use of the Differencing technique along with caching significantly reduces the data transmission rate over wireless communication links. Differencing prevents the repeated transmission of the same data on the low bandwidth wireless link. Thus, a mobile computing device can overcome the drawbacks of the slow, unreliable, and low bandwidth wireless communication link.

4. Protocol Reduction

The use of caching and differencing techniques can significantly reduce the amount of data that is transmitted over the wireless communication links. However, caching and differencing techniques do not address the overhead of repeated TCP/IP connections and redundant header transmissions. The WebExpress employs techniques to minimize the overhead of both repeated TCP/IP connections and redundant header transmissions. WebExpress is a software system that significantly reduces data volume and latency of wireless communications. WebExpress provides wireless communication reduction by using an interception technology transparent to Web users and servers.

Each mobile computing device connects to its home server with a single TCP/IP connection. All of the mobile user's requests are transferred over this single TCP/IP connection. Requests and responses are multiplexed over the connection.

The WebExpress system eliminates most of the opening closing connections by establishing a single TCP/IP connection between the mobile computing device and home server. The mobile computing device intercepts document requests and connections from the browser, and then sends them to the home server over a single TCP/IP connection. The mobile computing device's home server establishes a connection with the destination Web server for each request received from the mobile computing device, and then forwards the mobile user's requests to the destination server. Once the home server receives a response from the destination server, the connection with the destination server is closed. Then, the home server forwards the response to the browser and closes its TCP/IP connection with the browser. The connection setup and teardown overhead is incurred between browser-mobile computing device and the home server-Web server, but not between the mobile computing device and its home server.

WebExpress uses virtual sockets to provide that multiplexing support. Virtual sockets provide a mobile computing device to establish a single TCP/IP connection with its home server and use this connection for many HTTP requests. Virtual sockets permit efficient transport of HTTP requests and respond while maintaining correct HTTP protocol and WebExpress transparency for Web browsers and servers. [Ref. 42]

5. Header Reduction

When a mobile user establishes a connection with its home server, the mobile computing device sends its capabilities to its home server only on the first request. The mobile computing device capabilities are maintained by the home server during the connection.

The HTTP requires that each request contain the browser's capabilities. This information about the browsers' capabilities is the same for all requests for a given browser. The mobile computing device includes the capabilities as part of the HTTP request that are forwards to the home server.

HTTP requests and responses are prefixed with headers. HTTP request headers contain a list of content-types. This list informs the home server of the various document formats which the browser can handle. It is unnecessary to send this list across the wireless communication link in every request, because the list is constant for the browser. The mobile computing device sends that list to its home server in the first request while establishing a connection. This list is saved by the home server and the mobile computing device as part of the connection state information. The mobile computing device compares the list received from the browser with its saved version for each request. If they match, the list is deleted from the request before it is sent to the home server. When the home server receives a request from the mobile computing device with no access lists, it inserts its saved copy into the request header. If an access list is present in the received request, the home server replaces the saved version. Consequently, the correct access list is sent to the home server as if there were a direct browser-server connection.

HTTP response headers are normally different for each request. However, response headers vary only by a few bytes from one response to another. Encoding the constant content-type data can reduce the response to just a few bytes. This reduction can be worthwhile when multiplied by all the mobile computing devices sharing a wireless communication link.

6. Data Access Profile

A data access profile is the information that is of particular interest to the mobile user. Each mobile user has his own data access profile and sub-profile which are stored on the mobile user's home server. Upon login, the mobile user's data access profile and sub-profile are used to build the proxy and mobile computing device cache.

Profiles help to minimize wireless mobile computing communication cost. Because the mobile support station knows which data is of interest, and then it sends large data blocks to the mobile computing device at startup rather than demand paging. Because the cost of connection setup and teardown is high, sending fewer longer messages is more cost effective than sending many shorter messages. Downloading the mobile user's sub-profile to the mobile computing device cache reduces the number of requests by the mobile device. The dollar cost of sending information over a wireless network is high. Hence, fewer requests and replies are transmitted over the wireless link.

7. Delayed Write Mechanism

Delayed write mechanism is used to minimize the number of transfers from the mobile computing device to the mobile support station.

A mobile computing device batches its updates and then dispatches those updates periodically to the mobile support station. Batching updates result in fewer long messages rather than many small messages. In wireless mobile computing, the overhead of connection setup and teardown is high, so fewer longer conversations are better than many short conversations in terms of communication cost. Therefore, the delayed write technique helps a mobile computing device to conserve power and to minimize mobile communication cost by reducing the setup and teardown costs.

Delayed Write Mechanism is managed with the help of mobile computer's caching mechanism. Old data must be removed to provide room for new data in cache memory. Caches use a mechanism to choose least-recently-used data blocks for replacement. On average, blocks remain unreferenced for almost an hour before they are replaced. Usually, only clean blocks are replaced and dirty blocks are always written back to the mobile support station long before they can be replaced.

Dirty cache blocks can be written to the mobile support station for several reasons, and the most common is the delayed write policy. The delayed write policy forces blocks to be written to the mobile support station after they have been dirty for thirty seconds. The data integrity is the reason why dirty bytes leave the cache. Dirty blocks almost never leave the cache to make room for other blocks. They are usually written out to make new data permanent by the delayed write mechanism. Therefore, increasing the size of the file cache does not reduce write-back traffic. The write traffic can only be minimized by increasing the write-back delay.

8. Strict and Loose Reads

There are two levels of read consistency, strict-read and loose-read. Strict-read returns the most up-to-date information. Strict-read call returns the latest consistent value written by a strict reader by contacting all servers and finding and retrieving the most up-to-date copy. A strict-read is needed when a mobile user needs the most recent valid copy of a page and before any writes. A proxy is then responsible for retrieving the page either by requesting that page from the home server or recalling it from its own cache memory.

Loose-read allows the mobile user to access data which is available in the mobile computing device cache or in the proxy cache. Loose-read returns information -- the information is not necessarily the most up-to-date, but the most accessible. The loose-read technique reduces bandwidth requirements, because fewer requests are initiated from the mobile computing device. So, the mobile connectivity on wireless links can be enhanced in terms of performance and reliability.

9. Semantic Callbacks and Validators

Large communication latency over wireless links increases the cost of validating of cached pages, because maintaining cache coherence under wireless weak connectivity conditions is an expensive process. The cost of cache coherence is exacerbated in systems like Coda[2]. The Coda system uses anticipatory caching for availability. In anticipatory caching, the number of cached pages is much larger than the number of pages in current use. The

[2] Coda system is one of the earliest systems to demonstrate that an optimistic replica control strategy can be used for serious and practical mobile computing. Coda demonstrates that disconnected operation is feasible in a distributed Unix file system. Coda has shown that weak connectivity can be exploited to alleviate the limitations of disconnected operation.

Coda system maintains cache coherence at multiple levels of granularity and uses callbacks. Users and servers maintain version information on both individual objects and entire sub-trees. Cache validation is provided by comparing version stamps on the sub-trees, and validity is maintained through callbacks. The Coda system preserves accuracy while reducing the cost of cache coherence under weak connectivity conditions.

Satyanarayanan, M. recommends maintaining cache coherence at multiple granularities to a variety of data types and applications in the following ways [Ref. 2]:

- A mobile user caches data satisfying some predicate *P* from the server.
- The server remembers a predicate *Q*, which is much cheaper to compute, and possesses the property *Q* that implies *P*. In other words, as long as *Q* is true, the cached data is guaranteed to be valid. However, if *Q* is false, nothing can be inferred about that data.
- On each update, the server re-evaluates *Q*. If *Q* becomes false, the server notifies the user that its cached data might be stale.
- Before its next access, the user must contact the server and obtain fresh data satisfying *P*.

Satyanarayanan refers to *Q* as a semantic callback for *P*. The interpretation of *P* and *Q* depends on the specifics of the data and application. *Q* must conform to *P*: a simpler select statement in the first case, and a piece of code that performs a less accurate pattern match in the second case. In the Coda system, *P* corresponds to the version number of an object equal to a specific value *x*. *Q* corresponds to the version number of the encapsulating volume unchanged since the last time the version number of the object was confirmed to be *x*. [Ref. 2]

Semantic validation is valuable when the timing difference between local and remote actions is too large. The predicate *Q* serves as an inexpensive validator for cached data. Semantic Callbacks and Validators technique is valuable in mobile computing and widespread distributed systems, because preserving cache coherence under wireless weak connectivity conditions can be expensive and large communication latency can increase the cost of validation of cached objects.

10. Data Allocation

Wireless communication is more expensive than wire communication because of low bandwidth, and can become very expensive for mobile users that perform many accesses per day. Wireless communication can be minimized by using an appropriate data allocation scheme. If a mobile user frequently reads an object, and that object is not updated frequently, then it is beneficial for the mobile user to allocate a copy object to the mobile computing device. In this way, read operations access the local copy of the object and do not require communication. If the mobile user reads objects relatively infrequently as compared to the update rate, then a copy of the object should not be allocated to the mobile computing device.

If an allocation is warranted, then one of the two kinds of allocation schemes is used to allocate an object to the mobile computing device, the one-copy allocation scheme or two-copies allocation scheme. In the one-copy scheme, only the home server has a copy of the object. In the two-copies scheme, both the home server and the mobile computing device have a copy of the object.

An allocation method determines whether the allocation scheme changes over time or not. In a static allocation method, the allocation scheme does not change over time. In contrast, the allocation scheme changes over time in dynamic allocation method. There are two static allocation methods, one uses the one-copy scheme, and the other uses the two-copies scheme and a family of dynamic data allocation methods. These static allocation methods are used to select the allocation scheme according to the read/write ratio. If the reads are more frequent than the writes, then the two-copies allocation scheme is used; otherwise, the one-copy allocation scheme is used.

The family consists of all the methods that allocate and de-allocate a copy of an object to the mobile computing device based on a sliding window of requests. For every read or update, the latest requests are examined. If the number of reads is higher than the number of writes and the mobile computing device does not have a copy of object, then a copy of object is allocated to the mobile computing device. If the number of writes is higher than the number of reads and the mobile computing device has a copy of object, then the copy is de-allocated. Hence, the allocation scheme is dynamically adjusted according to the relative frequencies of reads and writes. Data allocation algorithms are implemented by software residing on both the mobile computing device and its home server. Data Allocation Method minimizes the amount of data transferred over wireless links to reduce the wireless communication cost.

11. Data Compression

Data compression consists of taking a stream of symbols and transforming them into codes. If the compression is effective, the resulting stream of codes will be smaller than the original symbols, so the amount of data transferred over wireless communication links are reduced.

Conventional data compression techniques achieve a thirty to fifty percent data reduction depending on the data. Wireless communication services that charge by the byte (rather than by the duration of the connection) use compression, but charge by the uncompressed byte. Hence, strategy to minimize cost is to perform the compression before sending the data to the wireless service.

Data compression techniques can be divided into two major parts, lossless compression and lossy compression. Lossless data compression reduces data size by eliminating the redundancies in the data. Lossless compression guarantees to generate a duplicate of the input data after a compress/expand cycle. Hence, each bit in the data can be restored precisely by decompression. Lossless compression is implemented using one of two different types of modeling: dictionary-based modeling and statistical modeling. Dictionary-based modeling uses an algorithm to replace strings of characters. As the dictionary-based algorithm reads an uncompressed data, it examines the data for recurring patterns. When the algorithm identifies a pattern, it writes the pattern to a dictionary. The dictionary is stored as part of the compressed data. The algorithm uses a shorter pointer that tells where the omitted pattern can be found in the dictionary. Statistical modeling reads in and encodes a single symbol at a time using the probability of that character's appearance.

Lossless data compression can be used when compressing databases or word processing files. The extent to which the data shrinks depends on the type of the data. Some types of data such as databases and word documents, are prone to redundancies and are particularly susceptible to compression. In these applications, the loss of even a single bit could be catastrophic.

Lossy compression is fundamentally different from lossless compression in one respect: lossless compression accepts a slight loss of data to facilitate compression. Lossy compressed files cannot be decompressed to their original state. Lossy compression is performed on analog data stored digitally. Lossy data compression is effective when applied to digitized voice and graphics images. Graphics files use lossy compression that reduces the file size by discarding data whose loss will not be noticed, such as small variations in color. A lossy compression program does not change the basic nature of the graphics images.

Consequently, data compression techniques achieve data reduction over wireless links. Wireless communication cost can be minimized performing data compression before sending the data to the wireless service that charges the mobile user by the byte.

12. Data Scheduling

There are two kinds of data broadcasting modes in mobile computing environments, one-way communication two-way communication modes. In the one-way communication mode, the server repeatedly broadcasts data on a public channel and mobile computing devices listen to that channel and access the data of interest in the broadcast stream. In the two-way communication mode, mobile computing devices send requests to the server and then receive replies from the server.

There are two kinds of parameters related to data broadcasting, access time and tuning time. Access time is the time that elapses from the moment a mobile user submits a request to the receipt of data of interest on a channel. Tuning time is the time spent by a mobile user listening to the channel.

The data scheduling method finds a broadcast schedule of data for minimizing the access time of the requests issued by mobile computing devices. Chung, Y. D., and Kim, M. H. introduce a wireless broadcast data scheduling method called QEM [Ref. 10]. QEM constructs the broadcast schedule by expanding the Query Distance Schedule (QDS) of each query after sorting the queries based on frequency. The algorithm used in this method is described in Figure 3.1, and its basic policies are as follows:

- **Policy 1:** Higher-frequency query takes precedence over the lower-frequency query when expanding the schedule.

- **Policy 2:** When expanding a query, that is its QDS, the Query Distance (QD)'s of the queries, previously expanded, remain unchanged.

- **Policy 3:** When expanding the QDS of query q_i into the currently constructed schedule, the proposed method always minimizes the QD of q_i, as much as possible.

```
Algorithm:  QEM
Input:      A set of data D and a set of queries Q
Output:     A broadcast schedule S
Method:     1. Initially S is empty.
            2. Sort the queries in decreasing order of $freq(q_i)$.
            3. For each query $q_i$ in the sorted order, expand S
               with $q_i$ by using the QDS Expanding Rules.
```

Figure 3.1: Description of the QEM Algorithm. [From Ref. 10]

The authors give the following example to introduce the QEM data scheduling method [Ref. 10]. They assume that there are eight data objects to be broadcasted and three queries that mobile users submit onto the broadcasting channel. The data set of each query is depicted as shown in Figure 3.2. The authors assume that all data objects are equal in size and also that the occurrence frequency of each query is freq (q_1)=3, freq (q_2)=2, and freq (q_3)=1.

Initially, the schedule S^{step0} is empty. According to Policy 1, the QEM algorithm finds the highest frequency query, q_1, and expands its QDS. Then the current schedule S^{step1} is formed as: S^{step1} = [d1, d2, d4, d5]. Second, the algorithm expands the query q_2, whose QDS is {d4, d5, d6, d7, d8}. Since the current schedule S^{step1} contains the data objects d4 and d5 that are in QDS (q_2), the schedule is expanded into one of these forms:

$S^{step2}{}_{RightAppend}$ = [d1, d2] [d4, d5] [d6, d7, d8]
$S^{step2}{}_{LeftAppend}$ = [d6, d7, d8] [d4, d5] [d1, d2].

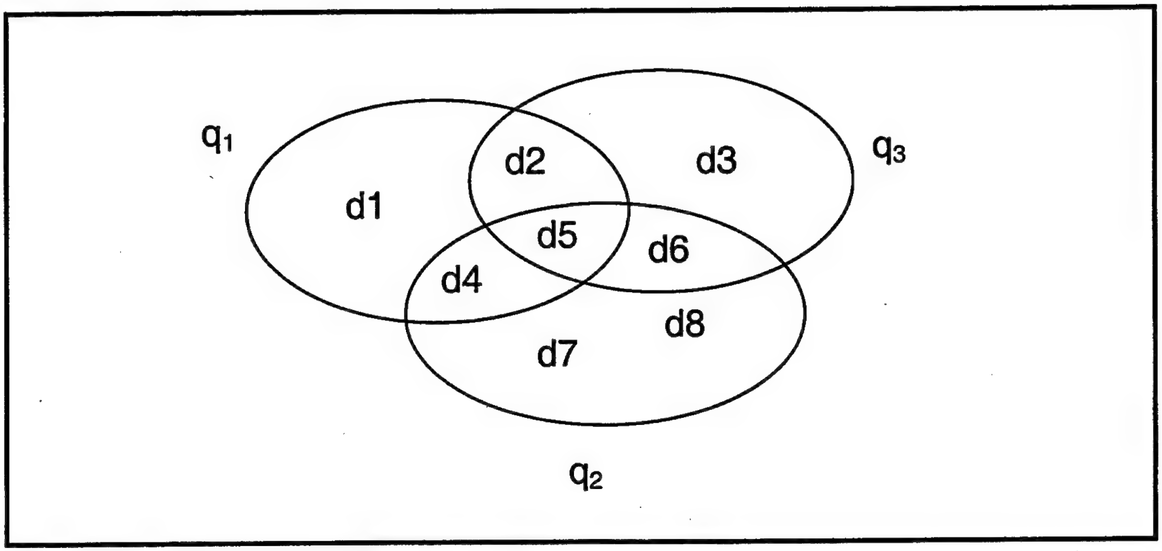

Figure 3.2: Queries and their QDS's. [From Ref. 10]

The former schedule is the result of appending QDS (q_2) at the right end of Sstep1, whereas the latest one is that of left appending.

As the data objects bounded by '[' and ']' are freely interchangeable, there are 24 (6*2*2) possible ways of data ordering for each schedule. The schedule S^{step2} minimizes the QD of q_2 (Policy 3) while preserving the QD of q_1 unchanged, that is QD (q_1, S^{step1}) = QD (q_1, S^{step2}) (Policy 2).

Finally, the QDS (q_3) is expanded. Among the data objects in QDS (q_3), only d3 is not included in the current schedule S^{step2}. Inserting d3 into the schedule increases the QD of q_1 and q_2, violating Policy 2. Hence, d3 must be appended to the left or right end of S^{step2}. When appending d3, the data objects of d2, d5, and d6 have to be moved for minimizing the QD of q_3 as follows:

$S^{step3}_{RightAppend}$ = [d1] [d2] [d4, d5] [d6, d7, d8] [d3]
$S^{step3}_{LeftAppend}$ = [d3] [d1, d2] [d4, d5] [d6] [d7, d8].

In the two schedules above, $S^{step3}_{LeftAppend}$ gives smaller TQD, for QD (q_3, $S^{step3}_{LeftAppend}$) is less than QD (q_3, $S^{step3}_{RightAppend}$) and those of q_1 and q_2 are equal. Thus, the final schedule will take one of the following forms that are results of $S^{step3}_{LeftAppend}$:

< d3, d1, d2, d4, d5, d6, d7, d8 >
or < d3, d2, d1, d4, d5, d6, d7, d8 >
or < d3, d1, d2, d5, d4, d6, d7, d8 >
or < d3, d2, d1, d4, d5, d6, d8, d7 > and so on.

The QEM method minimizes access time by the efficient scheduling of wireless broadcast data. QEM effectively constructs wireless broadcast schedules and yields a twenty percent reduction in access time [Ref. 10]. This reduction is significant in access time on low-bandwidth wireless links in terms of performance.

13. The Proxy Process

A mobile computing device transfers data of interest over the wireless communication links. Wireless links are slow because of the limited bandwidth and unreliable because of the frequent disconnections. Therefore, the requests of mobile user may time-out or several retries may be required. These limitations increase the amount of traffic over the wireless link. Fortunately, the amount of data traffic can be reduced by delegating the mobile user's tasks to the proxy process.

A proxy can be executed in a fixed location, or it may be mobile. Many systems that support wireless mobile computing provide proxies. Proxies perform various functions on behalf of their mobile users, and can be used to process control information or to manipulate mobile user information. A proxy can also be used for a mobile user to request a default connection, such as a voice connection to their home, or a data connection to their office.

The proxy process manages data and services for the mobile users. When a mobile computing device registers with a mobile support station, the proxy is created upon registration for the mobile user at the mobile support station. To obtain the mobile user's profile, the proxy contacts the mobile user's home server. The proxy caches the mobile user's profile and sends the sub-profile to the mobile computing device.

Proxies can be used to filter or modify application information being sent to a mobile computing device. For example, a wireless web browser may not have the bandwidth available to receive images embedded in a page, would be better served by receiving only text information [Ref. 34]. The amount of bandwidth used by the application on the air interface is greatly reduced by filtering at a proxy.

The proxy knows what the mobile computing device cache holds. Therefore, a proxy can filter out data that is sent by the home server and broadcast only data necessary to update the mobile computing device cache. This kind of data filtering minimizes the transfers from the mobile support station to the mobile computing device, and reduces the power consumption of the mobile computing device while receiving data.

The mobile computing device sends its requests to the proxy. The proxy retrieves pages on behalf of the mobile computing device. This process minimizes the number of retries and timeouts. Hence, the amount traffic over wireless link is reduced.

When the mobile computing device disconnects from the wireless link to conserve power, it cannot receive any messages. The proxy buffers messages and invalidations until the mobile computing device is ready to receive them. When the mobile computing device reconnects, the proxy sends updates to the mobile computing device. This feature helps save time, as otherwise, the mobile computing device cache needs to rebuild itself.

The proxy process is very useful for mobile computing. After the proxy is submitted, the mobile computing device can be disconnected from the wireless network. Hence, a proxy helps to save on wireless communication costs that can be very expensive over wireless links.

14. Adaptation Strategy

Adaptation insulates mobile users from the drawbacks of the mobile computing environments by using local resources to reduce wireless communication traffic. The needs of mobile users vary according to the capabilities of their computing devices, such as processing speed, screen size, etc. The features of the computing devices make it difficult for servers to provide an appropriate level of service to their users. Application-level adaptation solves these problems and provides mobile services to the users regardless of their computers' capabilities.

Fox, A., Gribble, S. D., Chawathe, Y., and Brewer, E. A., introduce a proxy-based approach to adaptation [Ref. 39]. In the proxy-based approach, the proxy agents reside between servers and users to perform users' tasks. The authors identify the following advantages of the proxy approach over the server-based [3] and user-based [4] approaches:

- Leveraging the installed infrastructure through incremental deployment. The enormous installed infrastructure and its attendant base of existing content is too valuable to waste; yet, some clients cannot handle certain data types effectively. User and network heterogeneity should allow interoperability with existing servers by enabling incremental deployment while evolving content formats and protocols are tuned and standardized for different target platforms. A proxy-based approach lends itself naturally to transparent incremental deployment, since an application-level proxy appears as a server to existing users and as a client to existing servers.

- Rapid prototyping. Software development on "Internet time" does not allow for long deployment cycles. Proxy-based adaptation provides a smooth path for rapid prototyping of new services, formats, and protocols that can be deployed to servers or user later if the prototypes succeed.

[3] Server-based approach attempts to insert adaptation machinery at each end server. [Ref. 39]
[4] User-based approach attempts to bring all users up to a least-common-denominator level of functionality. [Ref. 39]

- Economy of scale. A large central server is more efficient than a collection of smaller servers in terms of cost and utilization. Standalone desktop systems represent one server per user. This supports the argument for Network Computers, and suggests that co-locating proxy services with infrastructural elements (such as Internet points-of-presence) achieves effective economies. [Ref. 39]

a. *Taxonomy of Adaptation Strategies*

Satyanarayanan, M. introduces the taxonomy of adaptation strategies in [Ref. 2] as shown in Figure 3.3. The range of strategies for adaptation is delineated by two extremes, Laissez-faire (no system support) and Application-transparent (no changes to applications).

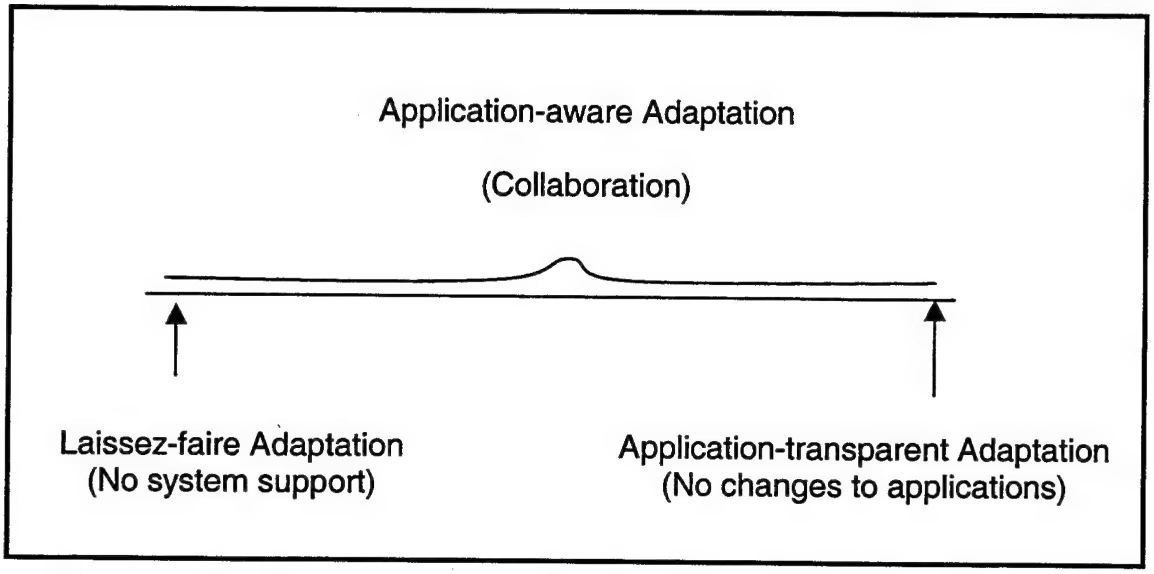
Application-aware Adaptation
(Collaboration)
Laissez-faire Adaptation
(No system support)
Application-transparent Adaptation
(No changes to applications)

Figure 3.3: Range of Adaptation Strategies. [From Ref. 2]

In Laissez-faire adaptation, system support is unnecessary and adaptation is the responsibility of individual applications. The Laissez-faire approach makes applications more difficult to write and fails to amortize the development cost of support for adaptation.

In an application-transparent adaptation, the system is responsible for adaptation. The system provides the focal point for resource arbitration and control. The application-transparent approach is attractive, because it is backward compatible with existing applications. Applications continue to work without any modifications when mobile.

Application-aware adaptation lies between Laissez-faire adaptation and Application-transparent adaptation. Application-aware adaptation permits applications to determine how best to adapt by preserving the ability of the system to monitor resources and enforce allocation decisions. [Ref. 2]

15. Resource Revocation

A mobile computing device owns and manages all of its system resources. A mobile computing device may revoke resources delegated to an application at any time.

Application-aware adaptation complicates the mobile computing device's resource management. Some applications may be more important than others so, the resource revocation strategy used by the mobile computing device must be sensitive to such variations. The cost of revocation of the same system resource may vary from application to application. For example, reducing the bandwidth for one application might result in increasing the amount of processing that it does to compensate. A similar reduction in bandwidth for another application might result in a smaller increase in processing. A good resource revocation strategy must take into account these differential impacts. There may be dependencies between processes that should be taken into account, because revoking resources from one process might cause another process to stall or cause deadlocks to occur.

Revocation of system resources from applications is uncommon in today's computer technology. Consequently, today there is little relevance resource revocation technique. [Ref. 2]

16. Auto Connect/Disconnect

The special software that resides on the mobile computing device can dynamically connect and disconnect from the wireless network. In disconnected operation, a mobile user continues to use data in the mobile computing device cache memory. Disconnecting from the wireless network when the mobile computing device modem is inactive has two important benefits. First, wireless mobile communication costs are reduced and the battery life of the mobile computing device is extended by avoiding wireless transmission and reception. Disconnected operation allows radio silence to be maintained -- an important capability in military applications.

Mirghafori, N., and Fontaine, A., identify and introduce the following states of auto connect and disconnect operation in [Ref. 3]:

- **Startup.** Startup is the initial powering up of the mobile computing device. The mobile computing device registers with a mobile support station upon entering the broadcast range of mobile support station. Upon registration, the mobile computing device notifies mobile support station of its home server address. The mobile support station creates the proxy process, which retrieves the mobile user's profile from its home server. The home server sends the pages in the mobile profile, marks the mobile computing device as a valid reader of those pages, and notes where to contact the mobile computing device. The proxy receives and caches the mobile user's profile, and then broadcasts the sub-profile to the mobile computing device. The mobile computing device receives and caches the sub-profile pages.

- **Sleep.** There are two types of sleep, voluntary sleep and involuntary sleep. Voluntary sleep is a planned power-down. In voluntary sleep, the mobile computing device cleans its dirty pager, gives up any write-locks it holds, and informs the proxy of its intention to sleep. Then, the proxy updates mobile computing device sleep time and buffers messages and invalidations for the mobile computing device until the mobile computing device wakes up and is ready to receive messages and invalidations. Involuntary sleep is an unplanned power-down, i.e., a sudden system crash. In an involuntary sleep case, the proxy does not know that the mobile computing device is not listening and continues to broadcast invalidations. The mobile computing device will recover missed messages upon wakeup by asking for all messages sent after its disconnection time. If the sleeping mobile computing device does not return, the mobile computing device state is sent to the home server after a system-specific amount of time and the proxy is killed.

- **Wakeup.** Wakeup is the powering up process after a mobile computing device has been asleep. Upon wakeup, the mobile computing device waits to get the mobile support station's address. Upon receiving the address, the mobile computing device sends a wakeup notification to the mobile support station and requests missed messages. In the case of voluntary sleep, the mobile computing device sends its sleep-time to the mobile support station. If the mobile computing device had gone to sleep involuntarily, the mobile support station proxy uses time- mobile computing device-contacted-proxy and time-invalidation-propagated for each page on its cache to calculate how many old messages should be resent.

- **Move/Handoff.** The mobile computing device listens for the mobile support station's address. When the mobile computing device notices that the mobile computing device is in a different region, the mobile computing device contacts the mobile support station. The mobile support station contacts the old mobile support station to obtain the state of the mobile computing device proxy. The old mobile support station cleans any dirty pages to the home server and sends the proxy state to the new mobile support station. The new mobile support station proxy contacts the home server to tell the home server where to contact the mobile computing device, and then broadcasts any invalidation to the mobile computing device. [Ref. 3]

17. Adaptive Antennas

In the past, antenna-related technology has typically received less attention from researchers as compared to the current wireless communication systems. Recently however, the focus of modern research is shifting towards adaptive (*smart*) antennas.

There are three main categories of adaptive antennas: switched beam, direction finding, and optimum combining. Tsoulos, G. V. identifies and defines these three adaptive antenna approaches [Ref. 15]. The switched beam method employs a grid of beams and chooses the beam, which gives the best signal-to-noise ratio. The direction finding method focuses on the acquisition and tracking of one parameter and the directions of the users. With the optimum combining method, the output signal-to-interference-plus-noise ratio is the parameter optimized. Tsoulos summarizes the most important advantages and disadvantages of these techniques as shown in Table 3.1.

Tsoulos makes one further categorization of smart antennas as shown in Figure 3.4: Spatial Filtering for Interference Reduction (Figure 3.4a) and Space Division Multiple Access (Figure 3.4b). The goal of spatial filtering for interference reduction (SFIR) is twofold. (1) Support one user in each of the co-channel cells of the reuse pattern employed, (2) interference reduction in the spatial domain to achieve a lower cell repeat pattern.

With space division multiple access (SDMA), an adaptive antenna system is deployed in such a way that multiple users within the same cell can operate on the same time (tk) and frequency (fk) channel by exploiting the spatial separation of the users. This concept is a dynamic sectorization approach in which each mobile computing device defines its own sector as it moves [Ref. 15].

Tsoulos summarizes the major advantages and disadvantages of these two methods in [Ref. 15] as shown in Table 3.2.

Approach	Advantages	Disadvantages
Switched beams	• Easily deployed • Tracking at beam switching rate	• Low gain between beams • Limited interference suppression • False locking with shadowing, interference and wide angular spread
Direction finding	• Tracking at angular change rate • No reference signal required ratio • Easier downlink beam-forming	• Lower overall carrier-to-interference gain • Susceptible to 10 signal inaccuracies; needs calibration • Concept is not applicable to small cell non-line of sight environments
Optimum Combining	• Optimum signal-to-interference-plus-noise ratio gain • No need for accurate calibration • Performs well even when the number of elements is smaller than the number of signals	• Difficult downlink beam-forming with frequency division duplex and fast time division duplex • Needs good reference signal for optimum performance • Requires high update rates

Table 3.1: Advantages and Disadvantages of Different Smart Antenna Approaches. [From Ref. 15]

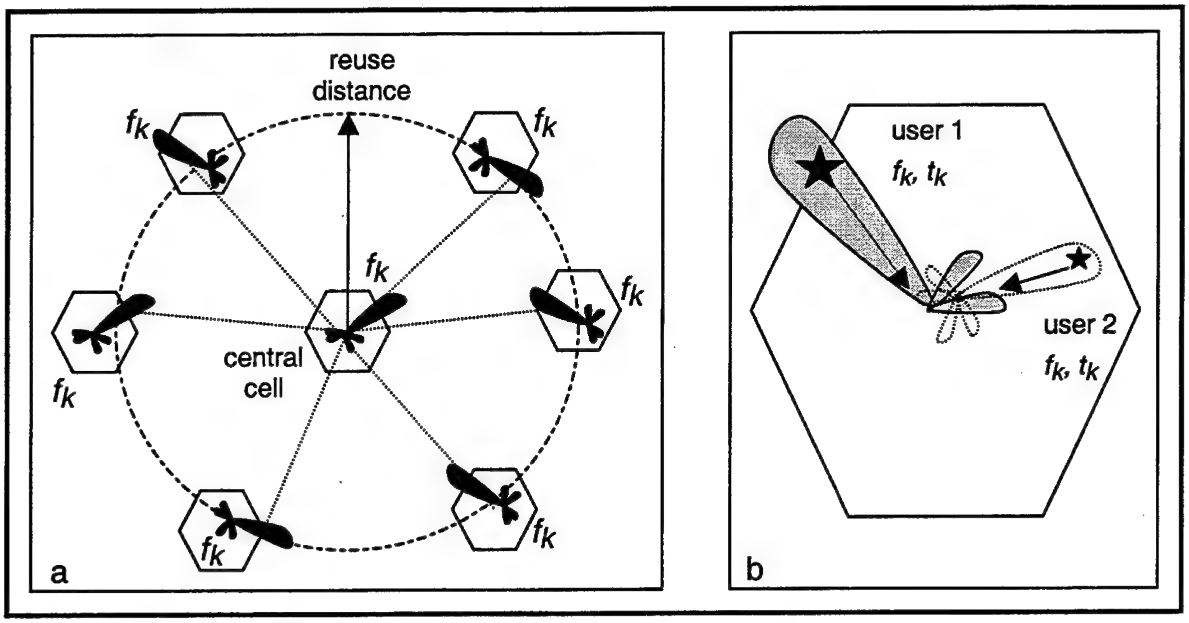

Figure 3.4 (a) Spatial filtering for interference reduction and (b) Space division multiple access concept. [From Ref. 15]

	Advantages	Disadvantages
Space division multiple access (SDMA)	- No need for revised frequency planning to exploit capacity gain - Single cell deployment for local capacity improvement	- Requires discrimination between intra-cell Space division multiple access users - More complex radio resource management (angle and power)
Spatial filtering for interference reduction (SIFR)	- No need for major air interface changes - Minor or no changes to the radio resource management	- Relies on intelligent intra-cell handover - Large deployments necessary to exploit the full capacity potentials

Table 3.2: Advantages and Disadvantages of Space Division Multiple Access (SDMA) and Spatial Filtering for Interference Reduction (SFIR) [From Ref. 15]

B. TRADE-OFFS BETWEEN COMMUNICATION MODELS

In a mobile computing environment, restrictive factors include limited battery power of the mobile computing device and the low-bandwidth of the wireless communication links. A mobile computing device communicates with its home server via wireless links, which are slow and unreliable. The power limitation leads to low-end mobile computing devices and the bandwidth limitation leads to higher-end mobile computing devices with caches.

Wireless communication can become very expensive for mobile users that perform many accesses per day. Wireless communication can be minimized by using an appropriate data-allocation scheme. If the mobile user reads objects infrequently compared to the write rate, then a copy of object should not be allocated to the mobile computing device cache memory. If a mobile user frequently reads an object and that object is updated infrequently, then a copy of that object is allocated to the mobile computing device cache. Later, the infrequent updates are transmitted from the home server. In this way, mobile user can perform read operations on the copy of object locally without wireless communication.

Caching minimizes wireless communication traffic, because mobile users perform read operations locally. Mobile users can access data more quickly if it is cached. A mobile user can cache its frequently accessed items into the mobile computing device and then perform read operations on its mobile computer locally. Hence, caching improves the performance and availability of frequently accessed items for query processing during disconnection. Caching combined with loose-read capability can reduce wireless traffic significantly, since mobile user's read operations are performed locally.

Differencing along with caching can also significantly reduce the data transmission rate over wireless communication links. In the differencing method, a frequently accessed item is cached on both the mobile computing device and its home server. When the home server receives a response from a web server, the home server computes the difference between the web server's response and the item that is in the mobile computing device cache. Then, the home sever sends the difference to the mobile computing device cache. The differencing technique prevents the repeated transmission of data on the wireless link.

Caching and differencing techniques can reduce the amount of data that is transmitted over the wireless link. However, these techniques do not address the overhead of repeated TCP/IP connections and redundant header transmissions. The WebExpress that is a software system can reduce the overhead of repeated TCP/IP connections and redundant header transmissions. The WebExpress system eliminates most of the overhead of opening and closing connections by establishing a single TCP/IP connection between the mobile computing device and its home server. In this manner, the WebExpress system reduces data volume and latency of wireless communications.

Large communication latency over the wireless communication links increases the cost of validation of cached pages. Maintaining cache coherence under wireless weak connectivity is an expensive process. The cost of cache coherence can be minimized by using Semantic Callbacks and Validators technique. Validity is maintained through semantic callbacks. Semantic callbacks and the validation technique are valuable when the timing difference between local and remote actions is too large. Therefore, this method is very useful in mobile computing and distributed systems.

The communication between a mobile computing device and its home server has to be kept at a minimum to achieve a reasonable response time.

Reasonable response time can be achieved through an extensive use of caching and the proxy process. The proxy process manages data and services for mobile users. When the mobile computing device sends its requests to the proxy, the proxy retrieves items of interest on behalf of the mobile computing device. The proxy knows what the mobile computing device cache holds. Therefore, a proxy can filter unnecessary data and broadcast only necessary data to update mobile computing device cache. This data filtering minimizes wireless communication between a mobile computing device and its home server, and reduces the power consumption of the mobile computing device while receiving data. Hence, the proxy process combined with a caching mechanism is one of the most significant means used to reduce wireless communication costs.

Data access profiles further enable the mobile user to customize cache and better allow the proxy to manage data. A data access profile is the information that is of particular interest to the mobile user. Downloading the mobile user's profile to the mobile computing device cache reduces the number of requests by the mobile user. Therefore, fewer requests and replies are transmitted over the wireless link. Hence, wireless communication cost is minimized, because the dollar cost of sending information over a wireless link is high.

Adaptation insulates mobile users from the drawbacks of the mobile computing environment by using local resources in order to reduce wireless communication traffic. The different features of the mobile computing devices, such as processor speed, memory etc., make it difficult for servers to provide optimum service to their users. Application-level adaptation provides services to the users regardless of the capabilities of the mobile user's computers.

The data scheduling method finds a data broadcast schedule for minimizing the access time needed for the requests issued by mobile computing devices. Access time is the time elapsed from the moment a mobile user submits a request to the receipt of data of interest on the communication channel. The data scheduling method reduces the access time by the efficient scheduling of wireless broadcast data. Therefore, communication between mobile computing device and its home server is minimized, and hence the communication cost is reduced.

In wireless computing environments, sending a few longer messages is cheaper than sending many small messages, because the overhead of connection setup and teardown is high. Delayed write mechanism is a useful method, which is used to minimize the number of transfers from the mobile computing device to its home server. Delayed write technique helps to conserve power and minimize mobile communication cost by reducing the setup and teardown costs.

If the wireless communication service charge their mobile users by the byte rather than by the duration of the connection, then data compression technique can be used to minimize communication cost. Data compression is performed before sending the data to the wireless service. If the data compression is effective, then the resulting stream will be smaller than the original data.

Disconnecting from the wireless link reduces the communication cost and extends power of the mobile computing device when the modem is inactive. In disconnected operation, a mobile user continues to use data of interest, which resides in cache memory. Auto connect/disconnect operation managed by special software helps saving on wireless communication cost.

Wireless communication costs can be further reduced by using adaptive antennas. Smart antennas will provide advantages that minimize wireless communication, such as needing fewer base stations, covering larger areas, and improving trunking efficiency.

IV. MOBILE CODE, SCRIPTS, AND MOBILE AGENTS

A. MOBILE CODE

Mobile code denotes the programs that travel on a heterogeneous network from host to host, and are executed on remote computers in order to accomplish goals. Mobile code can be written by anyone and executed on any computer that runs a web browser regardless of the operating system and hardware configuration of the computer. Java, Safe-Tcl, and TeleScript are the most popular languages for implementing and executing mobile code.

1. Benefits of Mobile Code

Mobile code technology is based on the principle of autonomy of application components. Low-reliable communication channels and low-bandwidth require new design methodologies for applications in a mobile setting. In networks, some regions are connected through conventional links while others are connected through wireless links. Therefore, the design of applications becomes complex. It is important to cope with frequent disconnections and avoid generating traffic over the low-bandwidth links. Mobile code overcomes these limitations by specifying complex computations that can be moved over a network.

Mobile code enables service customization. In conventional distributed systems, servers provide a fixed set of services a priori accessible through an interface that is unsuitable for user needs. A solution to this problem is to upgrade the server with new functionality, increasing both its size and complexity without decreasing flexibility. The ability to request the remote execution of code helps to increase server flexibility without affecting the size or complexity of the server.

Mobile code provides data management flexibility and protocol encapsulation. In conventional systems, when data is exchanged among components, each component owns the code describing the protocol to interpret the data correctly. Mobile code provides solutions that are more efficient. If protocols are only rarely modified and are loosely coupled with data, an application may download the code that implements a particular protocol only when the data involved in the computation needs a protocol unknown to the application. If protocols are tightly coupled with the data they accompany, components could exchange messages composed of both the data and the code needed to access and manage such data.

Finally, mobile code supports the maintenance and deployment phases of the software development process. In a distributed setting, the action of installing or updating an application on each computer must be performed locally with human intervention. Mobile code assists by providing automation for the installation and rebuilding process. Code in a mobile program can analyze the features of the local platform and perform the correct installation and configuration steps.

2. Programming Languages for Mobile Code

a. Java

Java is an object-oriented class-based programming language created by SUN Microsystems. Some of the more prominent features of Java are its ability to interpret of a pre-compiled code and perform class loading from another computer over the Internet. Java is a well-known and widespread programming language for mobile code. The applet model has been created to use Java for mobile code.

Java classes called applets are short application programs that can be automatically downloaded and executed while visiting a Web page containing applets. The applets allow for active presentation of information and interactive access to a server. An applet resides at the Internet server, and is sent to the user's computer upon request or requirement. Once sent to the user computer, an applet is executed and then discarded so that it does not clutter the user's computer. Downloading of applets, provided by the Web browser, can be regarded as a mechanism for supporting fetching of stand-alone code.

Java attempts to establish a secure computing environment by providing a layered approach to software mechanisms. Some safety features were added to eliminate pointer arithmetic, unrestricted casts, unions, operator overloading, and multiple inheritance. Exception handling has been added for the creation of robust applications. Arrays and strings are built-in with range check. Automatic memory management has been added to guarantee against pointer errors. Java provides threads and serialized methods for concurrency. Java uses packages that group a number of classes and interfaces. The Java library provides data structures, support for graphical user interfaces, and access to network communication.

Java is a safe language and guarantees that type and access rules are always respected enabling a low-level security policy. Most resources requiring dynamic access control, such as the file system, are controlled by a centralized security monitor. All security-related methods are declared final so that applications and applets are forced to use the appropriate code. Malicious applets could redefine the method in a subclass without this protection.

b. Safe-Tcl

Safe-Tcl is a procedural script language designed to be simple, portable, and powerful. Safe-Tcl is an extension of Tcl and is based on the Tcl programming language used to support active e-mail. In Safe-Tcl, there are no communication or mobility mechanisms at the language level. Mobility and communication must be achieved using external support, like e-mail. In active e-mail, messages may include some code to be executed when the recipient receives or reads the message.

In Tcl, every value is represented as a string for simplicity. Tcl is a safe language. There is no notion of pointers, unchecked array accesses, or casts. Most of the features of Safe-Tcl have been included in the latest version of the Tcl language, and a plug-in for the Netscape browser, allowing scripts to be included in Web pages, like Java Applets. The goal of the Safe-Tcl is to be a secure and safe language. The greatest concern in the design of Safe-Tcl is a safe graphical user interface. Typical applications for Safe-Tcl include advanced user dialogues for ordering and voting. The Safe-Tcl language is much smaller than other languages for mobile code and has a small footprint.

c. TeleScript

TeleScript is an object-oriented class-based language, conceived for the development of large distributed applications, and designed for network programming. TeleScript is intended to be a specialized language for communication.

There are two major types of processes in TeleScript known as places and agents. A place passes communications among agents. An agent is the central concept in TeleScript. An agent autonomously travels on the network on behalf of its owner. Mobile processes in TeleScript run in a separate domain and can only interact with the engine in which they run. TeleScript is used an electronic marketplace where users can launch their agents to search and reserve data over the network.

TeleScript includes a number of different security measures to ensure the authenticity of agents. Each agent comes with a particular authority or identification of its owner. A positive aspect of TeleScript is that it tries to deal with denial of service attacks. TeleScript agents have their own initiative to travel and are more powerful than Java Applets. However, these agents are more dangerous, because it is hard or impossible to control them once they have been launched. An important feature of the TeleScript agent is that the user does not have to be connected to the network while his is acting. When the user reconnects to the network, the agent finishes its job and returns to the user.

d. Limbo

Limbo is a safe imperative language that it is based on C, and includes additional declarations such as abstract data types, automatic memory management, first-class modules, and preemptive scheduled threads. Pointer arithmetic and casts are not supported by the Limbo language. Limbo provides a library of standard modules for secure and encrypted communication, graphics, and network communication. Limbo provides type-safe linking at the user level by way of the built-in support for dynamic linking of modules.

e. Obliq

Obliq is a dynamically typed, lexically scoped, object-based language, designed for distributed object-oriented computations. Because Obliq is dynamically typed, and type errors are caught by Obliq and propagated to the origin site. Obliq belongs to a class of object-oriented languages in which objects are created by copying existing objects (the prototypes). Any value can be transmitted between hosts, but objects are local to a site and are not considered as values. Object migration can be programmed with closure transmission, object copying, and aliasing.

No special provision for security is in Obliq at the time of writing Obliq language. Besides the basic use of scope to control what is exported, Obliq supports weak mobility using a mechanism for synchronous shipping of standalone code.

f. Objective Caml

Objective Caml is a functional language, originating from Caml, and has been used as a language for mobile code in the development of the MMM Web browser. Objective Caml includes imperative features, such as a class-based object system, references and assignment that are all integrated within a functional core. Objective Caml offers automatic type reconstruction, and a higher-order module system in which modules have signatures. Objective Caml includes support for concurrency through threads and class-based object orientation through typing discipline.

Objective Caml includes library support for dynamic linking of object files. The dynamic linking used for applets constrains the use of the primitives that are considered dangerous. The advantage of Objective Caml is that it supports several programming paradigms such as imperative, functional, and is object oriented.

3. Mobile Code Security

In the past, a computer user needed to use the File Transfer Protocol (FTP) in order to download an application over the Internet, both installing and executing the application while relying on his own understanding of the process. However, by using the World Wide Web, these tasks have become transparent to the user, as mobile codes such as Java Applets and scripts can be automatically downloaded and executed on the user's machine without the knowledge of the user. One drawback, although, is that the security concerns become especially strong in this environment, because someone's application is running on the user computer without the approval and knowledge of the user, and the presence of the downloaded mobile code makes the user's computer vulnerable to attack.

Hostile mobile codes perform hostile activities on the user's computer, and can harm the destination host while being executed there. Hostile mobile codes may limit the availability of the system resources, attack the integrity of the system, violate the privacy of the user, or merely annoy or inconvenience the user.

The following approaches have emerged to provide assurance against the hostile mobile codes:

- Firewalling
- Sandbox
- Code Signing

- Proof-Carrying Code

If these techniques are considered individually, none of them can provide a high level of protection for the user's system resources from the hostile mobile code attacks. For instance, code-singing and sandbox models are being used together. Combining these two approaches with a firewall mechanism provides higher security for the system resources.

a. *Firewalling*

The objective of firewalling is to protect the user from network-based threats and attacks by providing a single control point where security can be imposed. A firewall provides a blockade between an internal network and an external network that is either insecure or untrusted.

A firewall system consists of proxy servers and screening routers. The screening router applies a set of rules and filters incoming IP packets based on information that is available in packet headers. A proxy server runs on a firewall system in order to perform a TCP/IP function as a proxy on behalf of the user. When a user requests a Web page by using his Web browser, his request is sent to a proxy. The proxy takes the user's request and forwards it to the related server, then receives the requested Web page from that server. When the Web page is received, the proxy server parses the Web page in order to identify applet tags. For each identified tag, the proxy replaces the named applet with the name of a trusted graphics server applet stored to the user's Web browser. The proxy sends this modified page to the Web browser, and retrieves the named applet and modifies its byte code to use the graphics server in the user's browser for all input and output, for each identified applet tag.

Subsequently, the proxy server forwards the modified applet to the graphic servers of the browser.

The firewalling approach involves choosing whether to run mobile code on the user computer or not and deciding where the mobile code enters the user's domain. Running a firewall or Web proxy may be useful to identify and examine mobile codes, and decide whether or not to run mobile code to serve them to the user. The proxy prevents hostile mobile code to access the user's resources.

b. Sandbox Model

There are two types of applets that cause problems for networks: malicious applets and attack applets. Malicious applets cause inconveniences rather than an actual loss for the computer by monopolizing the resources of the computer. Attack applets are the most dangerous mobile codes and try to exploit software bugs in the user's computer.

A sandbox is an area of the computer in which the mobile codes are run in order to protect the system resources from viruses and Trojan horses that may attack system resources. The sandbox model provides for the restricting of access to the file system and controls the establishment of network connections. The sandbox security model constrains the privileges of mobile code executing within a Web browser, and prohibits untrusted mobile code from using any system resources of the user computer. The purpose of the sandbox model is to contain hostile mobile code in the user computer in such a way that mobile code cannot cause any problem for the executing environment.

Java applications are partitioned into two groups: trusted and untrusted applications. Every local application is considered to be trusted and every remote application, such as a Java Applet, is considered to be untrusted. Trusted applications have full access rights to the system resources such as the file system and the network, and run in an environment without any restrictions. However, untrusted applications run in a sandbox that allows them to access only certain system resources.

The most common implementation of a sandbox is in the Java interpreter inside Web browsers. The security manager, the class loader, and the verifier are three main components of the Java Interpreter. The class loader converts remote byte codes into data structures. Any class loaded from the external network requires an associated class loader. Thus, the only way to add remote classes to a machine is via the class loader.

The verifier performs static checking on the remote code before it is downloaded. The verifier checks that the remote code does not overflow or underflow the operand stack, and that the remote code does not illegally convert data types and does not use registers improperly.

The security manager provides flexible access to potentially dangerous system resources. Safe operations are always allowed, but harmful operations cause an exception. A Web browser developer or a system administrator can control an applet's access to computer resources by modifying the security manager.

An error in one of these three components causes a violation of the security policy -- the biggest problem of the Java sandbox.

c. Code Signing

In the code signing method, the user has a list of trusted entities. When a mobile code is received from an external network, the user verifies that mobile code was signed by an entity on that list. Mobile code is run on the user computer, if it is on the user's entity list.

JavaScript 1.2 includes the notion of code signing. The creator of a script can add a digital signature to it. A signed script is able to request expanded privileges, gaining access to restricted system resources.

d. Proof-Carrying Code

Proof-carrying code ensures the protection of the system resources. Proof-carrying code statically verifies a proof and checks mobile code to make sure that mobile code does not violate safety policies when the mobile code is loaded. After the proof is verified, the mobile code can run on the user computer. For example, it is possible to construct a proof that checks mobile codes whether or not mobile codes contain buffer overflows.

The major drawback is that the proofs are written by hand in assembly language. However, research is still underway to construct software tools for the automation of the proof-carrying code generation.

B. SCRIPTS

A script is a record that consists of a sequence of commands. A script can be executed as if it were an executable program. When a script is executed, the commands in the script file are executed one-by-one in order. Scripts are specifically modeled for representing stereotyped sequences of events and originally used in natural language processing. They are commonly used to customize or add interactivity to Web pages on the World Wide Web. Scripts control and coordinate collections of behaviors. A script is characterized by its event list. The events are treated as symbols, an event sequence becomes a string, and a script can be represented as a language consisting of a single string.

A script has five components: casual chain, main concept, places, actors, and props. A sequence of events in a script is called a causal chain. The causal chain serves as a default sequence and represents exceptions to a step in the sequence. The causal chain can have pointers to subscripts, which handle these exceptions. A script includes places where the causal chain is valid. Props are the objects used by the actors in order to follow the causal chain and attempt to meet the main concept. The props and the actors are variables, which are instantiated at run time.

1. Scripting Languages

A scripting language is a limited programming language designed to extend the capabilities of another application and to perform special or limited tasks. Scripting languages such as Tcl, Perl, Python, and Visual Basic assume that a collection of useful components already exist in other programming languages and are intended primarily for connecting components rather than writing applications. Scripting languages are generally used to extend the features of components, but are rarely used for data structures and complex algorithms. Scripting languages are referred to as system integration languages or *glue languages*, which allow rapid development of gluing-oriented applications.

System programming languages such as Java, Ada, and C++ are usually compiled, but scripting languages are interpreted. Interpretation provides rapid turnaround during development by eliminating compile times. The benefits of the scripting depend on the application.

Scripting languages have become integral components in system development because of a shift in the application mix toward gluing applications. For example, the growth of the Internet has popularized scripting languages. The Internet, serving as a gluing tool, does not create any new data. Rather it makes data accessible to a number of existing Web pages. A scripting language makes it possible for all the connected components to work together to perform Internet programming tasks. For instance, JavaScript is a popular scripting language for developing Web pages. Another example is the graphical user interfaces (GUIs) that are being used to glue together applications to make connections between a collection of graphical controls and the internal functions of the application. All of the best rapid development of GUIs is based on scripting languages, such as Tcl and Visual Basic.

a. JavaScript Language

JavaScript is a simple object-based scripting language that is interpreted by Web browsers. JavaScript has no classes or inheritance features. Instead, it uses user-defined and built-in extensible objects. The JavaScript programs are integrated with a Hypertext Markup Language (HTML) page, and a

Web browser interprets and executes the JavaScript code. A computer user can type text, press buttons, and perform calculations with the help of the JavaScript enabled Web pages.

JavaScript models the browser window and browser state information by providing an object-instance hierarchy. For instance, a navigator object provides information about the browser to a script, and a history object represents the browsing history in the browser window. JavaScript automatically creates an object-instance hierarchy of the HTML document elements when it is loaded by the browser. JavaScript uses dynamic binding to check object references at runtime. JavaScript is loosely typed, so variable data types are not declared.

JavaScript code is interpreted by Web browsers rather than compiled. This means that JavaScript does not need a compiler, but does need an interpreter. JavaScript code is embedded in HTML, and its execution depends on the availability of the interpreter. Simple applications can be created and can easily interact with HTML documents.

JavaScript provides the scriptwriter with compact pre-built tools that enhance the interactions between the users and an HTML page. These compact pre-built tools allow responses page navigation, form input, and other events. Responses to the user actions can be invoked by JavaScript without network transmissions. This is the major advantage of JavaScript over scripting languages. If the user's interactions with a Web page are processed on the user's computer with JavaScript, then excess Internet traffic is avoided.

2. Using Scripts in Mobile Computing

The main reason for using scripts in mobile computing is to delegate tasks from a mobile computer to a network resource. Scripts are task executors. Tasks are designed as Scripts by the mobile host and then sent to a mobile agent to strengthen the weak flow of mobile communication.

There are two kinds of scripts: Personal Script and Remote Importable Script (RIS). Personal Script is designed by the mobile user and sent to the wired network in order to delegate the user's tasks. Remote Importable Script is delivered by an external network server to personal scripts or other remote importable scripts. Remote Importable Scripts can be viewed as Java Applets imported from a server over a wired network. Thus, these scripts increase the functionalities of the Web browser.

3. The Advantages of Using Remote Importable Scripts

One of the greatest limitations of the mobile computing environment is the weak flow of information caused by low-bandwidth and frequent disconnections. Remote Importable Scripts minimize wireless communication by importing remote running scripts on the wired network, which reduces the use of the wireless link for a script import. Importing a remote script during the execution of a personal script allows a personal script to require the loading of a remote script. Then, it can receive results and eventually to continue executing.

Remote scripts are delivered by servers on demand during the execution time of a personal script. The mobile user's personal script must be sure to get the latest release of the remote script. A global update phase among servers ensures the consistency of a remote importable script.

Another advantage of importing a remote script allows the mobile user to have a description of a way to access a network resource. For instance, the remote script can describe the list of servers, which must be contacted.

4. Script Security

When a remote importable script is imported, untrusted executable code comes from an external server and it is integrated at the execution of the calling mobile user's personal script. This executable mobile code has to be interpreted and analyzed to prevent access to unauthorized data. Remote importable script has the right to access the data space contained in the mobile user's personal script; the remote importable script has the same read, write, and modification rights on the data as the personal script.

Thus, security issues arise and access control policy has to be checked during the execution of the personal script. Therefore, a remote script must have a restricted data space, and only data used as parameters, (i.e., read and write) must be accessible from a remote script for secure execution. Only data sent as execution parameters has to be readable and modifiable with respect to the given read and write rights. When a remote importable script is requested by a personal script, the remote importable script may call another remote importable script. This type of call should be prohibited, because the security property may not be transitive.

Security issues must be taken into account for remote importable scripts, which require the use of certification, payment, confidential data, etc. Security requirements change according to the type of the remote importable scripts. Security requirements depend upon the specific type of remote importable script is imported. A classification of remote importable scripts is needed so that the protocol can request from the remote script server and that the behavior of the remote script loader is defined to support the classification scheme.

5. Classification of Remote Importable Scripts

a. Protected and public remote importable scripts

A public script is sent to any user, but a protected script is sent to only authorized users. A pair of keys (K_c, K_d) is used. K_d is the private deciphering key and belongs to the agent. K_c is a public key. Consequently, the server sends the script enciphered by K_c. Then, the responsible object uses K_d that is available to the agent. In this case, the protected script is located on the agent's server cache memory and must not be stored in the cache memory of the mobile user.

b. General and personalized remote importable scripts

The contents of the general importable scripts are identical for all users. The contents of the personalized importable scripts are adapted to the user. Personalization can be achieved with the help of generation parameters that determine all phases of the script generation. A script called with generation parameters can be viewed as a personalized script, and a script called without generation parameters is considered a general script.

c. Paying remote importable scripts

During script delivery, a final transaction is required in order to obtain a paying script. If the transaction takes place after the purchase of an access right, an identification / authentication is required.

d. Certified remote importable scripts

Certified scripts provide a higher level of security, and authenticity is needed for a certified script. Authenticity verification can be carried out by the object in charge of the script import. A pair of enciphering-deciphering keys (K_c, K_d) is used for authenticity. K_c is the private key belonging to a desired script server. K_d is the deciphering public key. When a certified script is requested, the responsible object sends the request to the script server. The script server returns the desired certified script enciphered by K_c. The object in charge of the script import deciphers the script using the public deciphering key.

e. Node and terminal remote importable scripts

When a remote script import is requested, this script may require the import of other remote scripts. When a public remote importable script requests another certified script, the operation cannot be executed because of authenticity. If the script is a certified script, then the sender of the certified script is the trusted server. When a certified script requests another script from another server, the transitivity relationship in the entrusted domain is not always guaranteed. Therefore, two kinds of scripts are needed, namely node and terminal remote importable scripts [Ref. 17].

6. Using Cache Memory with Scripts

When a personal script requires a remote importable script, this remote script is imported from an external network, executed on the user computer, and then removed. However, sometimes a personal script may frequently request the same remote importable script. In this particular case, a caching mechanism can be used to minimize data transmission instead of re-importing the same remote script.

Some remote importable scripts may be commonly requested by personal scripts attached to different mobile agents. For the purpose of ease of accessibility, these remote scripts are stored in several cache memories on the same server. If a personal script needs a remote script, the personal script uses the cache manager located on the acceptor server. If a personal script requests a remote script from the cache manager, and this remote script is stored in the cache memory, a message is sent to the server to check if the cached script is consistent with the remote script. If so, the cache manager sends the script. Otherwise, the script delivered by the server is returned.

Guaranteed consistency duration can be associated with most of the remote scripts. The script manager uses this duration to determine whether or not the script stored in the cache memory is consistent. The script is removed from the cache memory when it is stale. No extra communication traffic is required within these consistency durations.

C. MOBILE AGENTS

An agent is active, autonomous, and acts on behalf of a mobile user or another agent. Agents provide a very critical service for the user by searching information, filtering and extracting data from the external networks.

An agent can simply sit in its own environment and communicate through conventional means, such as messaging and remote procedure calling. This kind of agent is called stationary agent, and executes only on the system on which it begins execution. For example, word processors that have spell-checking programs, which alert the computer user when the user types a misspelled word, have agents that correct the misspelling. As another example, an agent positioned in the stationary network may keep track of incoming emails, acknowledge receipt, and reply to messages. If a stationary agent needs data that is not available on its own environment, then the stationary agent interacts with an agent on another system by using a communication mechanism such as remote procedure calling.

However, a mobile agent is free to travel among the servers in the external networks. A mobile agent is a piece of software program associated with a mobile user, and accepts tasks from the user to overcome drawbacks of the mobile environment. A mobile agent is composed of code and data, and can navigate autonomously over the heterogeneous networks without the need for continuous interaction. A mobile agent can transport its code and attribute values (state) that help it determine what to do when it resumes execution at its destination.

Mobile agents are very useful for mobile computing by acting as an interface between the mobile user and the external networks. An external network server that has to transfer messages to the mobile user sends these messages to the associated mobile agent, and this mobile agent forwards the messages to the mobile user. This mobile agent is responsible for mobility management.

The mobile user has to disconnect his mobile computer frequently because of the weak energy autonomy of his mobile computer. Therefore, the mobile user is unreachable from external networks. A mobile user has to wait for the end of the data delivery to be able to disconnect his mobile computer when he requests data from an external server. A mobile agent acts as the mobile user's representative and is always connected to the related external server. Thus, the mobile agent is able to receive requested data even if the mobile user's computer is disconnected. After the mobile agent is submitted, the mobile user can be disconnected from the network. As a result, a mobile agent can also save on transmission costs that can be substantial over wireless links.

1. Benefits of Mobile Agents

Mobile agents can be customized according to the mobile users' needs, and then sent to the external network servers in order to take the advantage of wireless links. It is very convenient for a mobile user to delegate his tasks to a mobile agent, which will execute the user's tasks while the mobile user is disconnected from the external networks. The mobile agent returns to the mobile user's computer from its current location when the mobile user is connected to the external network. This feature of the mobile agents helps to reduce the mobile computer's power consumption and minimizes communication costs over the wireless communication links.

Mobile agents are useful for minimizing the flow of information between networks. Data should be processed locally instead of transferring over the network when huge amount of data stored at external network servers. Mobile agents minimize the network communication by allowing mobile users to package a conversation and transport it to an external network server where interactions take place locally. Mobile agents have a higher degree of survivability, because they transport both state and code encapsulated within their abstractions. It is possible for a mobile agent to leave a network and execute on another network, if a network partially fails. Mobile agents can adapt dynamically by sensing the execution environment and reacting independently to changes. Therefore, a mobile agent can maintain its optimal configuration for fulfilling the tasks that were assigned by the mobile user. Mobile agents are fault-tolerant. If a network server being shut down, all mobile agents executing on that server are warned and given time to transport themselves to another server to be able continue their operations.

2. Mobile Agent Concepts

Electronic marketplace requires a network that will let customers and providers of services find one another and transact business electronically. Mobile agents act on their user's behalf to search information, find the best airfares, and send and receive messages, so on.

A mainframe computer may function as a shopping center for a mobile agent, (i.e., a special place where an agent can purchase airline tickets). Each agent occupies a *place*, or a context in which an agent executes. An agent can move autonomously from one place to another occupying different places at different times. The typical place is permanently occupied by only one agent. This permanent agent represents the place and provides its service, (i.e., airline-ticketing agent provides information about flights, airfares, and sells airline tickets to the customers). Travel allows a mobile agent to obtain a service and return back to its starting place. For example, a user's agent may travel from the user computer to a special ticketing place to obtain airline tickets, and then the agent may travel to the user computer back to describe the ticket information.

Telescript language lets a computer package an agent, its attribute values and its code so that the agent can be transported to another computer. The agent decides when such transfer is required. If travel cannot take place, the agent declares an exception. If travel is possible, then the agent finds its next instruction to be executed at its destination. Thus, language reduces networking to a single instruction.

Meetings motivate agents to travel. Two mobile agents are allowed to meet if they are in the same place. A meeting lets mobile agents call one another's procedures in the same computer. An agent may travel to a place in an external server to meet the stationary agent of that server providing the service offered. If the meeting succeeds, the two agents are placed in contact with one another, and if meeting fails, the agents throw exceptions.

A connection allows the agents to exchange information at a distance. If the connection is made, the two agents are granted access to each other. A connection between two agents in different places is made for the benefit of human user interactions. For example, a mobile agent that travels in search of airline tickets may send an agent to the user computer asking for the seat preferences of the user.

The authority of an agent in the electronic world is the individual or organization in the physical world that it represents. Authority system lets one agent discern the authority of another. A server must know the authority of any procedure to control access to its files. This issue is very important for network security. The system verifies the authority of an agent whenever it travels from one region of the network to another, and may demand reliable cryptographic forms of proof to verify the authority of an agent. Checking and controlling the mobile agent authorities helps prevent viruses by denying agents that contain the characteristics of a virus.

Authorities can limit the capabilities of agents by assigning permits, or data that grants capabilities, to agents. A permit can grant the right to execute a certain instruction and grant the right to use a resource in a certain amount. Permits protect authorities by limiting the effects of malicious agents.

3. The Architecture of a Mobile Agent System

When-Shyen, E.C; Lien, Y.N.; Shu, H.T., and Liu, H. propose a mobile agent infrastructure to support mobile computing in their paper, "*Mobility and Management Support for Mobile Agents*" [Ref. 29].

The authors focus on agent management and network transport for the purposes of supporting mobile computing. Figure 4.1 represents the infrastructure of a mobile agent system.

When a mobile user submits a request for services, a user agent is created by a mobile supporting server. The mobile supporting server accepts the request from the mobile user and invokes an instance of the user agent on the mobile user's behalf to carry out the requests. Then, the user agent queries the directory server to obtain the location information of the agent server to determine where the requested services can be fulfilled. The directory server represents the resources and the services of the service agents. The agent server provides the resources needed for mobile agents to carry out their tasks. The management server holds the status of the mobile agents and responds to

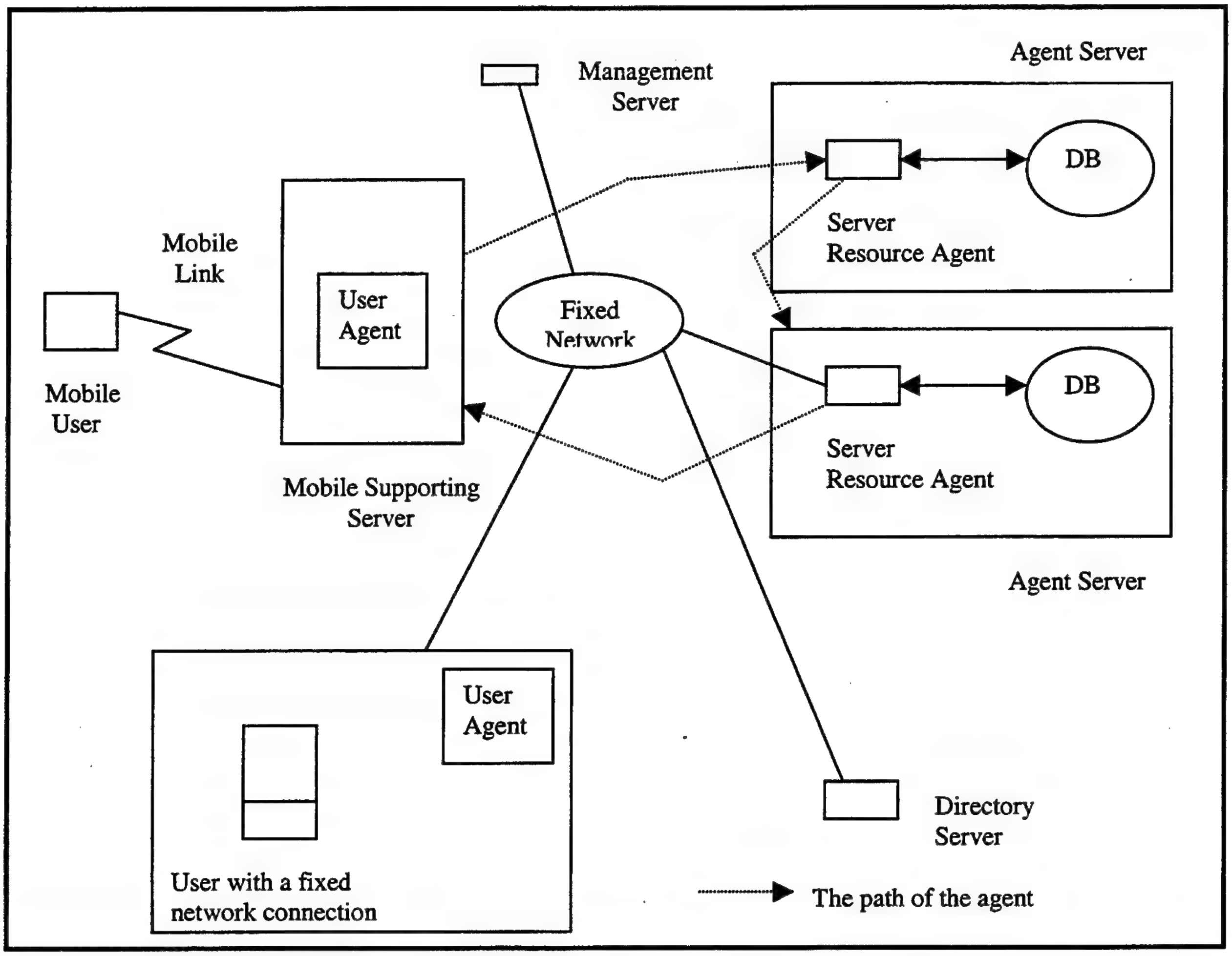

Figure 4.1: The Infrastructure of a Mobile Agent System. [From Ref. 29]

the queries in order to report the status of the agents. A service-resource agent is the stationary agent in the agent server where in the resources or services are provided.

Wong, D.; Paciorek, N.; and Moore, D. propose a generic java-based agent architecture in their paper "*Java-based mobile agents*" [Ref. 19]. Their mobile agent architecture consists of six components: an agent manager, a reliability manager, an application gateway, and a directory manager. (Figure 4.2 represents the generic mobile agent system architecture.)

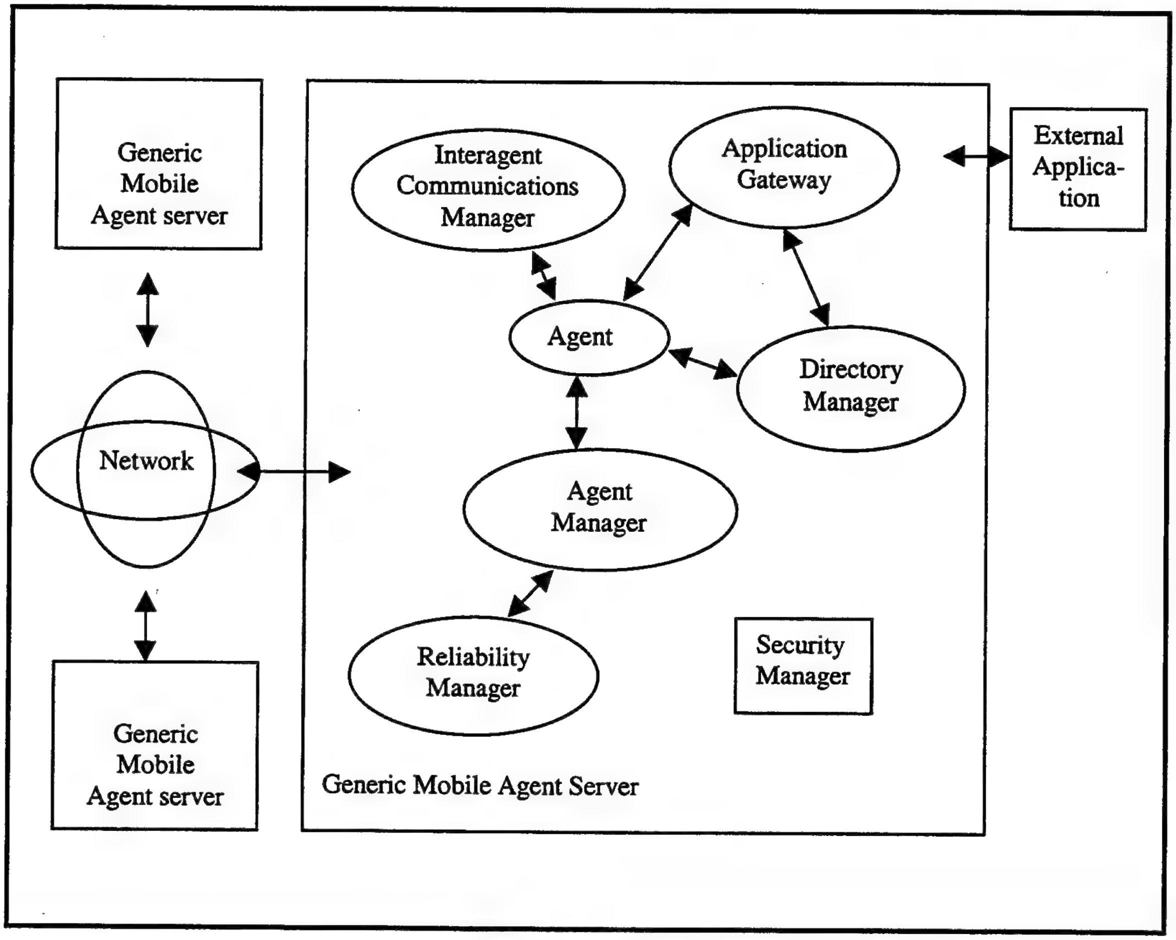

Figure 4.2: Generic Mobile Agent System Architecture. [From Ref. 19]

The agent manager receives agents from external networks for execution on the local server and sends mobile agents to remote servers. The agent manager serializes the agent and its state, and then passes the agent to the agent manager on the destination server. When the agent manager receives an agent, it reconstructs the agent, and then creates its execution context.

The security manager authenticates the agent before the agent's execution. The security manager protects the host and the mobile agents against unauthorized access. The security manager can also protect agents by encrypting them before transmission. The security manager also allows authorized agents to pass firewalls.

The reliability manager ensures the robustness of the mobile agent system, shields agents from the effects of the server, and guarantees the persistence of state associated with agents. The inter-agent communications manager facilitates communication between mobile agents through a network. The existence of multiple associate agents mandates inter-agent communication.

The application gateway serves as a secure entry point in which agents can interact with application servers. An arriving agent accesses resident servers through this gateway.

4. Languages for Mobile Agents

Language for the mobile agents should be flexible and able to adapt different circumstances, because an agent language must support the execution of flexible and safe software arriving from external networks. Although the language structure should be flexible, it must also set limits for mobile agents. For example, language should not allow the mobile agents to do anything that would endanger the user computer.

The stationary agent software can be written in C programming language. The stationary agents in servers allow interaction with local resources or services such as databases. A C program is compiled and linked with libraries into a set of executable code, which runs only on one particular hardware architecture.

Executable code takes control of the machine and may have complete access to all system services.

The mobile agents can be written in the Telescript and Java Languages. Telescript is the first agent development language, is interpreted, and has a security model similar to that of Java. Telescript language features many capabilities and desirable characteristics. For example, any algorithm can be expressed and an agent can be programmed to make decisions. Moreover, Telescript is safe, portable, and dynamic so that an agent can carry information among networks.

Use of Java is the most popular language for the development of mobile agents. Java is widespread, and the main features of Java include the interpretation of a pre-compiled code and class loading from an external network server. Mobile communications require that a mobile agent and its state be converted into a suitable form for network transmissions. Java's object serialization manages this conversion and reconstructs the mobile agent at the remote servers.

Java enables the migration of a mobile agent with the help of its class loading mechanism. The aim of the migration process is to reduce network bandwidth when multiple remote procedure calls are needed to execute an application. Java's class loaders dynamically load the classes included in an application through the network, and all of these classes are subject to security restrictions. These security restrictions help the mobile agent systems to protect their agents from unauthorized access.

Mobile agents, which are tightly integrated with the Web, can be developed by using Java. Applets can launch mobile agents from Web browsers and receive those mobile agents after they complete their tasks.

5. Mobile Agent Systems

Telescript is the first mobile agent system, and was created by General Magic for the development of large distributed applications. Telescript technology allows automated and interactive access to external network computers. Telescript technology supplements system-programming languages such as C or C++. The stationary software in user computers is written in C, which lets agents interact with users or system resources. The main purpose of Telescript technology is the electronic marketplace. Telescript opened the commercial marketplace for mobile agents and was followed by many mobile agent systems, such as Agent Tcl, Aglets, Voyager, etc.

Developed at Dartmouth University, **Agent Tcl** is one of the earliest systems on mobile agents. The Agent Tcl mobile agent system satisfies the needs of mobile computer users. Agent Tcl provides a Tcl interpreter, which is extended with support for strong mobility. It has extensive communication services, security mechanisms, and debugging and tracking tools. Agent Tcl protects individual machines against malicious agents, and provides support for docking station[5], enhances communication between agents, and minimizes wireless connection time.

The **Aglet model** was developed at the IBM Tokyo Research lab. The term aglet is word combining agent and applet. Aglet is the most widespread mobile agent system extending Java with support for weak mobility and it is available free. An aglet is a mobile Java object. It runs in its own thread of execution after arriving at an external network server.

The mobility of Aglets is classified in two ways, active and passive. An active aglet sends itself from its current location to a remote server, and a

[5] Docking system lets an agent jump off a connected computer such as laptop and return later.

passive aglet is pulled away by a remote server. An aglet can go to sleep temporarily and release its resources. This feature of an aglet helps to minimize the user computer's power consumption. Multiple aglets may exchange information to achieve a given task by the user. Yet, another advantage of the aglet is that it is easy to learn to use.

TACOMA is a joint project between Cornell University, USA, and Tromso University, Norway. In TACOMA, Tcl language is extended to include primitives that support weak mobility. TACOMA addresses the security and reliability issues of mobile agents. TACOMA mainly addresses operating system aspects of mobile agents rather than programming language aspects. Agents that execute units are implemented as Unix processes running the Tcl interpreter. The TACOMA system has been extended to support many interpreted languages, such as Python, Scheme, Perl, and C.

Ara (Agents for Remote Action), developed at the University of Kaiserslautern, is a mobile agent system that runs on Unix machines to support strong mobility, communication, and security. Mobility is supported through migration and data space management. Ara started with Tcl and C/C++ implementations, but currently also supports Java.

Mole, developed at Stuttgart University, Germany, in 1994, is one of the first mobile agent systems that was written in Java language. Mole is a Java API that supports weak mobility. Mole provides a stable environment for the development and usage of mobile agents. Mole addresses agent termination, groups of agents, and security for protecting mobile agents against malicious hosts. Mole agents are Java objects running as threads of the Java Virtual Machine.

Sumatra is a Java-based mobile agent system developed at the University of Maryland. Sumatra is an extension of the Java programming environment and supports adaptive mobile programs. Sumatra is based on a Java Virtual Machine, and facilitates the transparent migration of mobile agents. Mobile agents can suspend execution, migrate, and resume execution at the remote servers. Sumatra provides support for strong mobility of Java threads executed within execution engines. Threads can be migrated separately from the objects they need.

Voyager was developed by ObjectSpace and is a software package less than 300KB that supports mobile agents. Voyager is a Java-based system for developing distributed applications using mobile objects and agents. Voyager applications can be written once and then run on any platform that supports Java. Voyager has remote method invocation, object request broker, and support communication mechanisms. These mechanisms have made Voyager widely used claiming robustness and good performance.

MOA (Mobile Objects and Agents) is a mobile agent system recently developed at the Open Group Research Institute, and is written in Java. MOA contributes to resource management and transparent maintenance of communication channels across migration. It has been designed to support migration, communication and control of agent.

MASIF (The OMG Mobile Agent System Interoperability Facility) was developed by IBM, General magic, The Open Group, Crystaliz, and GMD Focus to standardize mobile agent system interoperability. MASIF addresses the interfaces between agent systems, but does not address the interfaces between agent applications and the agent system. MASIF standardizes interoperability between mobile agent systems by specifying agent management, transfer, and naming.

6. Mobile Agent Security

Mobile agent security is a critical issue for mobile users as it is difficult mobile agents to achieve. Current security technology is not good enough to support mobile agents.

Security in mobile agent systems can be identified as:

- The protection of a mobile agent from an attack by another malicious mobile agent, (inter-agent security),
- The protection of a mobile agent from an attack by a malicious external network server, (agent-host security),
- The protection of a host from an attack by malicious agents, (security between hosts and unauthorized third parties),
- The secure network transfer of agents, (inter-host security).

Existing cryptographic technology can be applied to provide inter-agent security, inter-host security, and security between hosts and unauthorized third parties. However, mobile agent-host security is specific to mobile code systems.

Hosts need to be protected against malicious mobile agents and mobile agents. The protection of hosts against malicious mobile agents can be managed by using the Sandbox security model. The Sandbox security model decides which programs can use system resources and which may not via special control components. The protection of mobile agents against malicious hosts is specific to mobile agents and is an open area of research.

a. Protecting the Host

The host can be protected against malicious mobile agents by using authentication, authorization, and enforcement. The host should be able to authenticate the owner of the mobile agent, and assign resource limits based on this authentication in order to prevent itself from denial-of-service attacks, theft, or damage to sensitive information.

Authentication verifies and authenticates the identity of the owner of a mobile agent with public-key cryptography. Each owner and its host has a public-private key pair. The host can authenticate the agent's owner if the mobile agent is digitally signed with the owner's public key. Or, if the mobile agent is digitally signed with the sending machine's key, the host will trust the sending machine.

After the identification of an agent's owner, the system assigns access restrictions to the mobile agents in a process called authorization. Then, the system ensures that the mobile agent does not violate these restrictions in the step known as enforcement. Authorization and enforcement assign resource limits to the mobile agent and enforce those resource limits. Resource limits include access rights, (i.e., reading a certain file, and maximum consumptions, i.e., total CPU time).

b. Protecting Mobile Agent

Protecting a mobile agent against a malicious host is a difficult security problem. There is no way to prevent a malicious host from examining or modifying any part of the mobile agents visiting the mobile host. The key challenge is to prevent the malicious host from using stolen information, and to detect tampering as soon as the mobile agent migrates onto the next host. There is no single mechanism that can solve this problem.

Currently, five-research directions exist for protecting mobile agents against malicious hosts:

(1) **The trust/reputation approach.** The trust/reputation approach allows mobile agents to transport themselves only to trusted hosts. Trust is a relationship between a mobile agent and a host. In this manner, a mobile agent knows that the host will not attack.

(2) **The organizational approach.** The organizational approach resolves the security problems by allowing only trusted parties to run mobile agents, and only trusted parties can operate mobile agents.

(3) **The manipulation detection approach.** The manipulation detection approach provides mechanisms to detect manipulations of the mobile agent code, enabling a mobile agent to detect and prove modification attacks.

(4) **The Blackbox protection approach.** The Blackbox protection approach attempts to generate a "black box" around the mobile agent that protects the mobile agent from read or manipulation attacks for a certain time interval. The mobile agent is considered a "*blackbox*," if the mobile agent's code and data cannot be read or modified. Only input to and output from the blackbox can be observed. If an agent completes the blackbox profile, the host cannot interfere with its execution in a directed way. Thus, an agent can be protected from host attacks. Currently, there is no known algorithm to fully provide blackbox protection.

(5) **Mobile Cryptography approach.** The Mobile Cryptography approach described by Sander and Tschudin [Ref. 49], uses encrypted programs to protect mobile agents from malicious hosts. Encrypted programs consist of operations that work on encrypted data. Malicious host cannot read or manipulate the mobile agent's original data unless it can break the data's encryption. The Mobile Cryptography approach has some advantages, such as

the protection of the mobile agent is easily provable, the protection is not time-limited unlike blackbox approach, and the cost of the protection is small.

All of these approaches are ongoing. None of them is used in real-world applications.

THIS PAGE INTENTIONALLY LEFT BLANK

V. USING MOBILE AGENTS IN MILITARY APPLICATIONS

Mobile agent technology leads to a new model of computing consisting of agents that are user enabled, embedded in an environment, dynamic in their behaviors, and able to improve their behaviors over time.

Intelligent agents are critical force multipliers for the military. Military personnel can simply manage difficult tasks with the help of the intelligent agents. Intelligent agents increase the productivity of military personnel by getting needed information, taking the action on the user's behalf, and deciding how to solve simple problems in a very short period of time.

This chapter introduces and defines some of the military projects that use mobile agents. The following projects represent the usage of mobile agents in military applications:

1. **Project Name:** Battlefield Intelligence Agents.
   Lockheed Martin (DARPA ITO sponsored Research).
   http://www.atl.external.lmco.com/projects/dais

This project develops and tests innovative uses of intelligent agent technology to facilitate intelligence collection and analysis in battles. The resulting technical developments of this project are expected to enable military intelligence units to exploit the rapidly expanding battlefield information sphere and demonstrate the viability of agent-based systems as intelligent middleware in distributed systems.

Wide Area Network (WAN) contains battlefield intelligence collectors, commanders, and data analysts. The objective of the project is to develop an intelligent agent system capable of locating and retrieving critical information in time to affect a battle. Mobile agents are used for data discovery within the wide area network.

Mobile users can connect to the battlefield is wide area network by using hand-held computers and radio links. Data of interest can reach a mobile user in minutes from its initial entry point with the help of the mobile agents. These capabilities allow highly mobile units such as counter intelligence teams to gain access to the information resources of the WAN, and enable them to launch a mobile agent onto the network, disconnect and relocate, then reconnect to find the agent waiting with its task results.

2. **Project Name:** Interaction Society Collaborative Agents.
Honeywell Technology Center (DARPA ITO Sponsored Research)
http://www.htc.honeywell.com/projects/hci

The goal of the project is to design and develop an architecture of software agents able to collaborate with human users for the performance of demanding and time-critical tasks. The approach to the development of the Interaction Society involves the creation of a new scalable protocol, involving the organization of reactive agents into social groups to constraint their behavior. This new protocol is embodied in the Search and Rescue Assistant System (SARA) and demonstrated in the military Search and Rescue (SAR) domain.

SARA supports military search and rescue operations by providing mixed initiative task assistance and situation awareness to search and rescue center operators. Reactive software agents that bound into societies to organize their behaviors act as collaborative partners with humans in information intensive tasks. Agents generate, manage, and create data visualizations, interfaces, and information to simplify human-computer interactions.

3. **Project Name:** Execution Monitoring Tool
http://www.darpa.mil/iso/jlactd/execumon.htm

The Execution Monitoring Tool provides pro-active assistants for tracking assets and an automated alert notification via electronic mail, cellular phone, pager, etc. Intelligent agents are employed to search for data in disparate databases in a manner that is transparent to the user.

The Execution Monitoring Tool allows a mobile user to create a list of the most critical items with which he is concerned. The intelligent agent searches for those items, gains knowledge about their location, status, and arrival times, and notifies the mobile user. The user can log onto a web page to check the status, or simply to have the agent e-mail, page, or call the user to inform.

4. **Project Name:** LCS Marine
(DARPA ITO Sponsored Research)
http://www.atl.external.lmco.com/projects/LCS-Marine/index.html

LCS Marine team is developing a prototype system that will enable computational entities to listen for information requests, compute user-centered solutions, and show tailored visualizations to individual war fighters in their operational domains. LCS technology will be integrated with intelligent agent technologies to revolutionize the way computers are used on the battlefield.

In the LCS Marine environment, individuals will explicitly define their information needs through an interactive dialogue with their local computing device. Spoken Language Systems software and intelligent agents will interact to identify the explicit and implicit tasks required. The mobile agents interact with specific information servers over tactical communication links to retrieve the required information.

5. **Project Name:** Mobile Agents for Network Trust (MANET)
(DARPA ITO Sponsored Research)
http://www.opengroup.org/RI/darpa/1997/f255Summ.html

The developers of the MANET project are developing a novel trust and authorization model, distributed infrastructure, and tools for responding to intrusions and attacks in a system of systems environment such as the Internet. The major components of the project are a trust model, an implementation of the model for mobile agents, and demonstration that the resulting infrastructure can respond to intrusions and attacks.

Mobile code is fundamental to attack detection and response in a system of systems environment, because it is impossible to predict the nature of future attacks. Mobile agents support adaptability in responding to attacks by allowing decisions about mobile code usage to be made at the point of use. The implementation of their trust model for mobile agents will be based on public key cryptography, innovative concepts of group membership, and ongoing work on the Adage authorization system.

6. **Project Name:** Foundations and Support for Survivable Systems (Cornell University) (DARPA ITO Sponsored Research)
http://www.cs.cornell.edu/Info/People/fbs/Arpa.DIW96.smry.html

In this project, researchers are seeking to develop technology to support survivable critical infrastructure systems with the following techniques: leveraging formal methods, investigating how cryptographic abstractions can be used in implementing fault-tolerance, and exploring the use of mobile code.

The researchers are investigating the agent paradigm for use in critical infrastructure systems. They are specifically augmenting their TACOMA system so that its agents can be fault-tolerant and immune to various attacks. In this project, mobile agents are used to structure a critical infrastructure system. A mobile agent facilitates maintaining site integrity by narrowing the interface between a computation and the sites visited.

7. **Project Name:** Quality-Based Reliable Computing (QBRC) (DARPA ITO Sponsored Research)
http://www.crhc.uiuc.edu/PERFORM/QBRC.html

Military applications require highly reliable information technology. This project will develop an approach to performing rapid diagnoses and recovery, and modeling and analysis tools that check whether the desired qualities are present to insure high quality information. The key to their approach is the use of a unifying measure for the quality of information presented to a decision maker as a measure of that information's dependability. In this project, active monitoring is employed to measure the quality of information presented to a decision maker, and guide system reconfiguration when the quality of the information presented becomes unacceptable.

Intelligent agents provide system monitoring by maintaining a current picture of system assets (data, software, and hardware) and their associated qualities. Intelligent agent architecture is scalable across both the distributed heterogeneous network and monitoring granularity. The researchers are developing three types of agents: system-mapping agents, base –quality-collection agents, and intrusion and fault detection agents. System mapping agents determine the network topology, hardware, software, and data configuration of the distributed heterogeneous network. Base-quality-collection agents assess the current quality state of the system assets while the intrusion and fault detection agents monitor the system and alert the Quality Propagation Specification (QPS) of any detected faults or intrusions.

VI. CONCLUSIONS & RECOMMENDATIONS

1. Conclusion:

There is an increased interest in mobile computing because of the growth in cellular communications. Mobile computers that easily connect to the Internet are becoming indispensable tools for a broad spectrum of society. Today, not only high-end computer users but also lay people use mobile computers for their daily tasks. Mobile users need to use their computers while traveling or while they are away from their working environments.

Mobile computing allows computer users to take their mobile computing devices such as laptops, palmtops, personal digital assistants, and other portable computers, away from their workplaces and to use them while traveling. Mobile computing gives users the opportunity to work with web servers and other network resources via wireless communication links from almost anywhere on the world. Mobile agents provide additional support by enabling movement of the programming environment along with the mobile computer.

As technology evolves, there will be more support for mobility in the underlying infrastructure of mobile computing environment, which makes mobility easier to deploy. For example, cellular phone services provide increased support for mobile computing.

Recommendation:

Mobile computers communicate with network servers over wireless communication links that are slow and unreliable. The communication between a mobile computer and a network server via a wireless link should proceed as if there had been no mobility. The infrastructure of the mobile computing environment should support the movement of mobile computers from one location to another. For example, mobile computers can be reconnected to a network server at a new location, and mobile agents can continue to communicate as they navigate on the Internet. Mobile agents can move to external networks very far away from the mobile computer in order to complete the mobile user's tasks. Therefore, the infrastructure of mobile computing environment should address performance, scalability, and reliability issues.

2. Conclusion:

Security is a critical issue and one of the biggest challenges facing mobile computer users. The existing security infrastructure is designed to protect stationary computers. Stationary computing systems such as network servers are more secure than mobile computing systems. Stationary computers can be protected by firewalls; however, mobile computers are much more difficult to protect.

In today's world of computing, mobile computers are vulnerable to attacks such as denial of service, Trojan horses, and viruses. For example, it is relatively easy to access data sent from a mobile computer and thus violate the integrity of a mobile agent by modifying its code or data while that agent is visiting a remote network resource.

Recommendation:

When a mobile computer restarts at a new location, the mobile computer's identity should be verified as some external network servers will need to trust the mobile computer's identity before communication can be established.

Mobile users do not trust foreign computers or mobile agents. Mobile users are often reluctant to provide foreign mobile agents access to local resources. If remote access is forbidden because of security concerns, a combination of mobile agents and controlled local access can be acceptable solution. On the other hand, it could be more difficult to attack a mobile agent because its location is not always known.

Security is hard to achieve for mobile agents, as current security technology is not strong enough to support mobile agents. Secure communication over nonsecure channels can be accomplished by encryption, which can be done in software or hardware. Existing cryptographic technology can be applied to provide inter-agent security, inter-host security, and security between hosts and third parties. The protection of hosts against malicious mobile agents can be managed by using authentication, authorization, and enforcement. The protection of mobile agents against malicious hosts is hard to ensure and is currently an open area of research.

3. Conclusion:

Mobile users need to be able to communicate with larger and more powerful machines via wireless LANs or cellular phones for optimal use of mobile computing systems. The designers of mobile computing systems must take into account the drawbacks of global communication such as propagation delay.

One constraint imposed by global computing is propagation delay. Wide area networks suffer from the problem of propagation-delay. If a program makes frequent use of remote data, its performance will suffer because of propagation delays. This problem may not be resolved anytime soon.

Recommendation:

Process migration can help alleviate propagation delay. In process migration, the mobile user's program is moved closer to the data of interest. First, a program starts running on the user's mobile computer. If the running program later needs to make frequent accesses to remote data, the mobile user's program is moved to a stationary computer, which is physically closer to the data. This method of process migration reduces the propagation delay. At the end of the program execution, the mobile user's program returns to his computer to display results. Finally, a program may complete its tasks while minimizing communication costs with the help of process migration.

4. Conclusion:

The World Wide Web is one of the most highly utilized applications in mobile computing, and the mobile user wants to access and use the Web at high speeds. The mobile user can access the Web easily over wireless links, but more slowly than over wired links. Wireless communication links are slow compared to wired links.

Recommendation:

Caching plays a key role in mobile computing when the overhead of using the wireless link increases. Effective caching can significantly reduce wireless link traffic. Mobile users can access data more quickly if it is cached, because mobile users perform read operations locally. Caching of data in a mobile computer minimizes the need for message transmission of data and improves the mobile computer's performance and the availability of services.

5. Conclusion:

Mobile computers do not have continuous network connection and they are often disconnected for long periods. Wireless network connections are often prone to sudden failures. The mobile computer can cease to function during the loss of one or more network connections. Frequent disconnections on wireless communication links are very common, so these connection failures are of great concern in mobile computing.

Recommendation:

The more autonomous a mobile computer, the better it can tolerate the loss of network connections. It is important for a mobile computing device to operate as a stand-alone computer in environments with frequent disconnections, because the mobile computing device can reduce wireless communication by running applications locally, rather than splitting the application across the wireless network.

Loss of network connections can be hidden by operating asynchronously, and using caching and delayed write techniques. In asynchronous operation, the mobile user sends multiple requests before asking for acknowledgement rather than waiting for a reply after each request. It is important to cache whole files on the mobile computer so that the entire file can be read locally during a network disconnection. These techniques reduce wireless communication and mask some network failures. Not all network disconnections can be masked; however, good user interfaces can provide some feedback about which operations are unavailable due to network disconnections.

In addition to the techniques listed above, mobile agents can also assist mobile computers while they are experiencing unexpected network disconnections. A mobile agent can move from one computer to another, suspend its execution at any point, transport itself to a new computer, and restart execution on the new computer. When its tasks are finished, a mobile agent returns to its starting point and displays the results. Consequently, the mobile user's tasks can continue even if the mobile computer experiences a loss of network connection.

6. Conclusion:

Weak flow of information over wireless links is one of the major constraints of mobile computing. Mobile users that have access their data on a low bandwidth wireless network experience weak data flow, and high error rates. Therefore, low bandwidth on wireless links is one of the key concerns with mobile computing.

Recommendation:

Mobile computing designs should place more emphasis on communication bandwidth constraints than on stationary computing designs because wireless communication links provide lower bandwidth than wired links.

Wireless communication bandwidth is divided among mobile users sharing the same cell. The usable bandwidth for each mobile user depends on the size and distribution of the mobile user population. More wireless cells can be installed to service mobile users thereby improving network capacity.

System performance can be improved by scheduling communication using techniques such as differencing, delayed write mechanism, data allocation, data compression, and data scheduling. Use of scheduling techniques help cope with the low bandwidth, and while they do not increase low bandwidth, they improve user satisfaction.

7. Conclusion:

Weak energy autonomy due to the limited battery power of mobile computing device is one of the major constraints of mobile computing. Battery weight is important for mobility because mobile users may have to carry spare batteries or recharge them frequently. Small batteries are easy to carry, but they can undermine mobility, because they provide less power. Therefore, power consumption of mobile computing device becomes a very important issue. Minimizing power consumption can improve mobility by reducing battery weight.

Recommendation:

Power consumption can be minimized by using faster processors, trading-off more sending messages for listening, or using power management software programs.

In order to retain more power in the battery, processors that can process at higher frequencies can be used to perform more work on each clock cycle.

Power can also be conserved by the efficient operation of both the user and the computer itself. Some power management software programs can control the mobile computer's management of auto connect and auto disconnect operations over wireless links. Some software programs can turn off screen lighting when not in use for a period. Further research can be done on screen lighting, because mobile computer screens consume a large amount of battery power.

Wireless transmission requires more power than wireless reception. Therefore, trading sending messages for more listening can also save battery power.

8. Conclusion:

Mobile computers have a higher risk of physical damage, loss, theft, and unauthorized access. These constraints may lead to a partial or total loss of the mobile user's data and/or privacy.

Recommendation:

The following recommendations can reduce the risks related with mobile computer use mentioned above:

- The risks can be reduced by minimizing the storage of essential data on the mobile computer. A mobile computer should serve only as a portable computing device in order to reduce the probability of data loss.

- Important data stored on disks and removable memory cards can be encrypted to help prevent unauthorized access. An authentication system can be used to login to the mobile computer.

- In order to prevent data loss, mobile user can keep a copy of data that does not reside on the mobile computer. One solution could be to have backup copies of important files.

- In the case of mobile computer theft, a special software system that resides on the mobile computer can help locating the mobile computer with the help of the Global Positioning System (GPS). GPS can locate the mobile computer's position with the help of the satellites.

9. Conclusion:

Mobile agents do not directly reduce wireless communication costs because they do not increase low wireless communication bandwidth. Mobile agents minimize substantial wireless communication costs by providing two types of reductions, the amount of data flow over the slow wireless network and the number of interactions between entities residing at external network computers.

Recommendation:

Mobile agent can minimize the amount of data transferred from the external network servers to the mobile computer by carrying a special code that provides data filtering. Mobile agents can execute data filtering processes on the external computers before the data is sent over the wireless network to the mobile computer. Consequently, excess data flow on the wireless link can be minimized with the help of mobile agents.

A reduction in the number of interactions between programs residing at external network computers can be achieved by bringing two programs to the same computing environment. Mobile agents are able to move the user's processes to remote computers, and can also move the data residing at the remote computer to the mobile computer. Consequently, the amount of data transferred over wireless links can be reduced with the help of mobile agents.

10. Conclusion:

A computer network may fail while a mobile agent is executing its owner's tasks on that network. When a network failure occurs, or a network completely goes down, mobile agents on that network are lost and cannot continue to execute objects. If a mobile agent is lost, the user's tasks cannot continue. Therefore, mobile agent's tasks must be saved in the case of a network failure.

Recommendation:

One solution to prevent the loss of mobile agents would be to create another mobile agent at the previous visiting network computer capable of managing the same tasks whenever the original mobile agent moves to a new network computer. This method can save mobile agent tasks in the case of a network failure. The mobile agent can continue to manage its tasks by restarting at the previous network computer. However, implementing this method is complex and hard to achieve.

An alternative solution would be for the mobile agent to create checkpoints in order to record its states during execution. Checkpoints could be saved at the computer, which is on the other network. Later, checkpoints could be used to restore the mobile agent and its internal state.

APPENDIX A. GLOSSARY OF TERMS

Access Time: is the time elapsed from the moment a client submits a query to the receipt of the data of his interest on the broadcast channel. [Ref. 10]

Agent: **1.** A program specifically designated to interact with a server and access data on the user's behalf. **2.** An automatic program that is designated to operate on the user's behalf performs a specific function in the background. When the agent has achieved its goal, it reports to the user. In the future, agents may roam the world's computer networks, looking for information, and reporting only when the information has been retrieved. [Ref. 31]

Applet: In Java, a mini program embedded in a web document that when downloaded, is executed by the browser. [Ref. 31]

Application: Application is a program that enables you to do something useful with the computer such as writing or accounting (as opposed to utilities, programs that help you maintain the computer). [Ref. 31]

Bandwidth: The amount of data that can be transmitted via a given communications channel (such as a computer network) in a given unit of time (generally one second). For digital devices, bandwidth is measured in bits per second (bps). The bandwidth of analog device is measured in cycles per second (cps). [Ref. 31]

Cache: A buffer storage that contains frequently accessed instructions and data; it is used to reduce access time. [Ref. 32]

Caching: Storing instructions and data in a cache. [Ref. 32]

CGI: Common Gate Interface. On the World Wide Web, CGI is the protocol that describes the standard method of communications between a web server and an external software application. CGI is used by the network as a means of handling the data from one computer to another. [Ref. 33]

Compute: to perform a calculation, for example, to add, subtract, multiply, or divide numbers. [Ref. 33]

E-commerce: The use of the Internet for business-to-business and business-to-consumer transactions. E-commerce is made possible by encryption technologies such as SSL. [Ref. 31]

Ethernet: A local area network hardware, communication, and cabling standard, originally developed by Xerox Corporation, that can link up to 1024 nodes in a bus network. A high-speed standard using a base band (single channel) communication technique, Ethernet provides for a raw data transfer rate of 10 megabits per second, with actual throughput in the range of 2 to 3 megabits per second. Ethernet uses carrier sense multiple access with collision detection (CSMA/CD) techniques to prevent network failures when two devices try to access to network at the same time. [Ref. 31]

Fault Tolerance: The capability of a computer system to cope with internal hardware and software problems without interrupting the system's performance, often by automatically bringing backup systems online when the system detects a failure. Fault tolerance is indispensable whenever computers are assigned critical functions, such as guiding an aircraft to a safe landing or ensuring a steady flow of medicines to a patient. Fault tolerance is also beneficial for non-critical applications. [Ref. 31]

Firewall: In a local area network or on the Internet, hardware and software through which all-incoming data must pass for the purpose of verification and authentication. If the security procedures were not satisfied, then unauthorized access would be denied. The term comes from the practice in the building profession of constructing firewalls between apartment complexes so that if a fire were to start in one block, it would not easily spread to another. Firewalls can also help system administrators to track computer usage and alter the access in the event of a security breach. They provide encryption where necessary and can protect certain areas of a network. [Ref. 33]

FTP: File Transfer Protocol. FTP is an Internet standard for the exchange of files. FTP is a specific set of rules that comprise a file transfer protocol. [Ref. 31]

Granularity: Granularity is the ability to deliver distinct digital data to finer and finer physical locations – ultimately and individual office or desk.

GUI: Graphical User Interface.

Heterogeneous Network: A computer network that includes computers and devices from several manufacturers and transmits data using more than one communications protocol. [Ref. 31]

Host: In a computer network, a computer that provides end users with services such as computation and database access and that may perform network control functions. [Ref. 33]

HTML: Hypertext Markup Language. The universal codes used for the World Wide Web to instruct a Web browser how a document is to be managed and displayed, and in particular, where the hypertext links will take the user. [Ref. 33]

HTTP: Hypertext Transfer Protocol. On the World Wide Web, HTTP is the file transfer protocol that enables the user to send and retrieve files across the Internet. HTTP allows the author of a web page to embed hyperlinks to other websites. [Ref. 33]

Internet: An enormous and rapidly growing system of linked computer networks, worldwide in scope, which facilitates data communication services such as remote login, file transfer, electronic mail, the World Wide Web, and newsgroups. Relying on TCP/IP, the Internet assigns every connected computer a unique Internet address, also called an IP address, so that any two connected computers can locate each other on the network and exchange data. [Ref. 31]

Killer Application: Industry jargons for a computer application that suddenly becomes so wildly popular that it drives other sectors of the industry. A killer application, for example, is an application that surpasses (i.e., kills) its competitors. Examples are VisiCalc, the original spreadsheet program, and the Web browser, which changed computing by putting a graphical face on the Internet and making it simpler to use. [Ref. 35]

Latency: In a computer network, the amount of time required for a message to travel from the sending to the receiving computer. This is far from instantaneous in a packer-switching network, given the fact that the message must be read and passed on by several routers before it reaches its destination and results in jitter. [Ref. 31]

LAN (Local Area network): Personal and other computers within a limited area that are linked by high-performance cables so that users can exchange information, share peripherals, and draw on programs and data stored in a dedicated computer called a file server. Ranging tremendously in size and complexity, LANs may link only a few personal computers to an expensive, shared peripheral, such as a laser printer. More complex systems use file servers and allow users to communicate with each other via e-mail to share multi-user programs and to access shared databases. [Ref. 31]

Mainframe: Mainframe is a multi-user computer designed to meet computing needs of a large organization. [Ref. 31]

Modem: A device that converts the digital signals generated by the serial port to the modulated analog signals required for transmission over a telephone line, and likewise, transforms incoming analog signals to their digital equivalents. Modems come in various speeds and use various modulation protocols. The most recent standard, called V.90 enables communication at 56Kbps. [Ref. 31]

Object-oriented: Conforming to the philosophy of object-oriented programming, in which programs are made up of interacting objects, which are self-contained, reusable program modules that support a specific function. Every object belongs to a class of generalized objects that all share the same function; by means of inheritance, objects within the class can automatically take on the class functions. A programmer can quickly create a new object by taking an existing abstract object of a certain class and filling in specific data and procedures as needed. [Ref. 31]

PC: Abbreviation for personal computer. In practice, this abbreviation usually refers to IBM or IBM-compatible personal computers, as opposed to Macintoshes. [Ref. 31]

PDA: Personal Digital Assistant; a term coined in 1992 by John Sculley (then Apple's Chairman) to describe hand-held electronic computerized products that can be used to assist users with telecommunications and messaging. Apple's first PDA was the Newton Message Pad. The term PDA is now used to refer to any device capable of assisting its user to become better organized in terms of administrative and telecommunications tasks. [Ref. 33]

Protocol: A strict set of rules that govern the exchange of information between computer devices. To communicate successfully, the communicating computers must use the same protocol. [Ref. 33]

Proxy: Also called proxy server. A program that stands between an internal network and the external Internet intercepts requests for information. A proxy is generally part of a broader solution to internal network security called a firewall. The purpose of a proxy is to prevent external users from directly accessing resources inside the internal network, or, indeed, knowing precisely where those resources are located. The proxy intercepts an external request for information, determines whether the request can be fulfilled, and passes on the request to an interval server, the address of which is not disclosed to the external client. By disguising the real location of the server that actually houses the requested information, the proxy makes it much more difficult for computer criminals to exploit potential security holes in servers and related applications, which might enable them to gain unauthorized access to the interval network. This protection from outside attack comes at the price of imposing inconveniences including configuration hassles and slower performance on internal users who wish to access the external Internet. [Ref. 31]

Real-time: The immediate processing of input, such as a point-of-sale transaction or a measurement performed by an analog laboratory device. The computers that are used in cars are real-time systems. [Ref. 31]

RPC: Remote Procedure Call.

Script: A series of instructions, similar to a macro and typed in plain text, that tells a program how to perform a specific procedure, such as logging on to an e-mail system. Some programs have built-in script capabilities. Some programs write the script automatically by recording your key-strokes and command choices as your perform the procedure. [Ref. 31]

Security: The protection of valuable assets stored on computer systems or transmitted via computer networks. Computer security involves the following conceptually differentiated areas:

- Authentication (ensuring that users are induced the persons they claim to be),
- Access Control (ensuring that users access only those resources and services that they are entitled to access),
- Confidentiality (ensuring that transmitted or stored data is not examined by unauthorized persons),
- Integrity (ensuring that transmitted or stored data is not altered by unauthorized persons in a way that is not detectable by authorized persons),
- Nonrepudiation (ensuring that qualified users are not denied access to services that they legitimately expect to receive, and that originators of messages cannot deny that they in fact sent a given message.[Ref. 31]

SQL: Acronym for Structured Query Language. In database systems, an IBM-developed query language that has become the standard for querying databases in a client/server network. The four basic commands (SELECT, UPDATE, DELETE, AND INSERT) correspond to the four basic functions of data manipulation (data retrieval, data modification, data deletion, and data insertion, respectively). SQL queries approximate the structure of an English natural language query. [Ref. 31]

Synchronous: Occurring together thanks to regular pulses received by some type of timing device. [Ref. 31]

TCP: Transmission Control Protocol. In TCP/IP, the set of standards (protocols) that enables data files to be sent in a reliable, error-free way from one computer to anther across the Internet or wide area network. The main TCP uses a transmission method known as PAR (positive acknowledgment with retransmission) whereby the computer that sends the data will continue to send the files to the specified destination until the receiver sends a signal to the sender to confirm that the data was received without error. [Ref. 33]

Tuning time: is the amount time spent by a client listening to the channel. [Ref. 10]

URL: Uniform Resource Locator. In the World Wide Web, the unique address of the Web site on the Internet. It allows the web browser to identify which file in which directory needs to be retrieved for the user. [Ref. 33]

Web Browser: A software program enables a user to access files from any computer that is connected to the Internet. [Ref. 33]

Web Server: In the World Wide Web, the software program that receives, manages, and responds to the requests for documents and files. The requests are structured using the Hypertext Transfer Protocol, and once processed, are sent back to the requesting software, which is usually a browser. [Ref. 33]

Workstation: In a local area network, a desktop computer that runs application programs and serves as an access point to the network. [Ref. 31]

LIST OF REFERENCES

1. Larry Francis, "Mobile Computing: A Fact in Your Future," paper presented at the 15th annual international conference on Computer documentation, pp. 63-67, 1997.

2. Satyanarayanan, M., "Fundamental Challenges in Mobile Computing," paper presented at the 15th annual ACM symposium on Principles of distributed computing, pp. 1-7, 1996.

3. Mirghafori, N.; Fontaine, A., "A Design for File Access in Mobile Environment," paper appears in Mobile Computing Systems and Applications, pp. 57-62, 8-9 December 1994.

4. Prasad Sistla, A.; Wolfson, O.; Yixiu Huang, "Minimization of Communication Cost Through Caching in Mobile Environments," paper appears in Parallel and Distributed Systems, IEEE Transactions on, pp. 378-390, April 1998.

5. Kravets, R.; Krishan, P., "Power Management Techniques for Mobile Communication," paper presented at the 4th annual ACM/IEEE international conference on Mobile computing and networking, pp. 157-168, 1998.

6. Lauzac, S. W.; Chrysanthis, P. K.; Tjoa, A. M.; Wagner, R.R., "Programming Views for Mobile Database Clients," this paper appears in Database and Expert Systems Applications, proceedings, 9th International Workshop on, pp. 408-413, 26-28 August 1998.

7. Chan B. Y.; Si, A; Leong, H. V., "Cache Management for Mobile Databases: Design and Evaluation," this paper appears in Data Engineering,

proceedings, 14th International Conference on, pp. 54-63, 23-27 February 1998.

8. Wolfson, O.; Yixiu Huang, "Competitive Analysis of Caching in Distributed Databases," paper appears in Parallel and Distributed Systems IEEE Transactions on, pp. 391-409, April 1998.

9. Barbara, D.; Imieli, T., "Sleepers and Workaholics: Caching Strategies in Mobile Environments," ACM SIGMOD Record, v. 23, No. 2, pp. 1-12, June 1994.

10. Yon Dohn Chung; Myoung Ho Kim, "QEM: A Scheduling Method for Wireless Broadcast Data," paper appears in Database Systems for Advanced Applications, Proceedings., 6th International Conference on, pp. 135-142, 19-21 April 1999.

11. Flinn, J,; Satyanarayanan, M., "PowerScope: A Tool for Profiling the Energy Usage of Mobile Applications, "paper appears in Mobile Computing Systems and Applications, Proceedings., Second IEEE Workshop on, pp. 2-10, 25-26 February 1999.

12. Barbara, D., "Mobile Computing and Databases-a Survey," paper appears in Knowledge and Data Engineering, IEEE Transactions on, pp. 108-117, January-February 1999.

13. Chakrabarti, S.; Dutta, G., "A Low Deviation Digital Modulation Scheme for Mobile Communication," paper appears in Personal Wireless Communication, IEEE International Conference on, pp. 193-197, 17-19 February 1999.

14. Goyal, A.; Sundareshan, M. K., "Performance Analysis of a Person-based Mobility Management Scheme for PCN," paper appears in Performance, Computing and Communications Conference, IEEE International, pp. 97-103, 10-12 February 1999.

15. Tsoulos, G. V., "Smart Antennas for Mobile Communication Systems: Benefits and Challenges," paper appears in Electronics & Communication Engineering Journal, pp. 84-94, April 1999.

16. Bhagwat, P.; Bisdikian, C.; Korpeoglu, I.; Krisha, A.; Naghshineh, M., "System Design Issues for Low-Power, Low-Cost Short Range Wireless Networking," paper appears in Personal Wireless Communication, IEEE International Conference on, pp. 264-268, 17-19 February 1999.

17. Carlier, D.; Trane, P., "Task Delegation Model Assigned to Mobile Computing," paper appears in Information, Communications and Signal Processing, 1997. ICICS., Proceedings of 1997 International Conference on, v. 1, pp. 220-224, 9-12 September 1997.

18. Lange, D.B.; Oshima, M., "Seven Good Reasons for Mobile Agents," Communications of the ACM, v. 42, No. 3, pp. 88-89, March 1999.

19. Wong, D.; Paciorek, N.; Moore, D., "Java-based Mobile Agents," Communications of the ACM, v. 42, No. 3, pp. 92-ff, March 1999.

20. Thorn, T., "Programming Languages for Mobile Code," ACM Computing Surveys, v. 29, No. 3, pp. 213-239, September 1997.

21. Fuggetta, A.; Picco, G. P.; Vigna, G., "Understanding Code Mobility," paper appears in Software Engineering, IEEE Transactions on, pp. 342-361, May 1998.

22. Schaefer, M.; Pinsky, S.; Dean, D.; Li Gong; Roskind, J.; Fox, B., "Ensuring Assurance in Mobile Computing," paper appears in Security and Privacy, 1997. Proceedings., 1997 IEEE Symposium on, pp. 114-118, 4-7 May 1997.

23. Hohl, F., "Mobile Agent Security and Reliability," papers appears in Software Reliability Engineering, 1998. Proceedings., The 9th International Symposium on, pp. 181, 4-7 November 1998.

24. Vogler, H.; Kunkelmann, T.; Moschgath, M.-L., "An Approach for Mobile Agent Security and Fault Tolerance Using Distributed Transactions," paper appears in Parallel and Distributed Systems, 1997. Proceedings., 1997 International Conference on, pp. 268-274, 10-13 December 1997.

25. Rubin, A. D.; Geer, D. E., "Mobile Code Security," paper appears in IEEE Internet Computing, pp. 30-34, November-December 1998.

26. Ghosh, A. K., "On Certifying Mobile Code for Secure Applications," paper appears in Software Reliability Engineering, 1998. Proceedings. The 9th international Symposium on, pp. 381, 4-7 November 1998.

27. Sander, T., "Protecting Mobile Code," paper appears in Software Reliability Engineering, 1998. Proceedings. The 9th international Symposium on, pp. 183, 4-7 November 1998.

28. Gritzalis, S.; Iliadis, J., "Addressing Security Issues in Programming Languages for Mobile Code," paper appears in Database and Expert Systems

Applications, 1998. Proceedings. The 9th International Workshop on, pp. 288-293, 26-28 August 1998.

29. Wen-Shyen E. Chen; Su, S. T.; Lien Y.-N.; Shu, H.T.; Liu, H., "Mobility and Management Support for Mobile Agents (poster)," paper appears in Proceedings of the 2nd international conference on Autonomous agents, pp. 451-452, 1998.

30. Neuenhofen, K.; Thompson, M., "A Secure Marketplace for Mobile Java Agents," paper appears in Proceedings of the 2nd international conference on Autonomous agents, pp. 212-218, 1998.

31. Pfaffenberg, B., *Webster's New World Dictionary of Computer Terms,* 7th edition, MacMillan General reference, A Simon&Schuster MacMillan Company, 1999.

32. McDaniel, G., *IBM Dictionary of Computing,* 10th edition, International Business Machines Corporation, August 1993.

33. Nader, J. C., *Prentice Hall's Illustrated Dictionary of Computing,* 3rd edition, Printice Hall Corp., 1998.

34. Thomas, F. L. P.; Krishan, K. S.; Richard, D. G., "Challenges for Nomadic Computing: Mobility Management and Wireless Communications," mobile Networks and Applications, v. 1, No. 1, pp. 3-16, August 1996.

35. *The Illustrated Book of Terms and Technologies,* 2nd edition, Computing Dictionary by PCNOVICE/Smart Computing, fall 1999.

36. Cohen, A. M., *A Guide to Networking*, An international Thomson Publishing Company, 1995.

37. Koy, A. S., "Assets On the Go," *Mobile Computing & Communication*, v. 10, No: 10, pp. 67-67, October 1999.

38. Milojicic, D., Douglis, F., and Wheeler, R., *Mobility Processes, Computers, and Agents, ACM*, 1999.

39. Fox, A., Gribble, S. D., Chawathe, Y., and Brewer, E. A., "Adapting to Network and Client Variation Using Active Proxies: Lessons and Perspectives," IEEE personal Communications, 5(4): 10-19, August 1998.

40. Baker, M., Hartman, J., Kupfer, M., Shirrif, K., and Ousterhout, J., "Measurements of a Distributed File System," Proceedings of the 13th ACM Symposium on Operating System Principles, pp. 198-212, October 1991.

41. Nelson, M., Gailly, J. L., *The Data Compression Book,* pp. 1-12, M&T Books, 1996.

42. Housel, B.C., and Lindquist, D.B, "WebExpress: A System for Optimizing Web Browsing in a Wireless Environment," *Proceedings of the Second Annual international Conference on mobile Computing and Networking*, pp. 108-116, November 1996.

43. Oppliger, R., "Internet Security: Firewalls and Beyond," paper presented at the Communications of the ACM, v.40, pp. 92-102, 1997.

44. Wallach, D.S., Balfanz, D., Dean, D., and Felten, E.W., “Extensible Security Architectures for Java,” paper presented at the sixteenth ACM symposium on Operating systems principles, pp. 116-128, 5-8 October 1997.

45. Giuri, L., “Role-based Access Control in Java,” paper represented at the third ACM workshop on Role-based access control, pp. 91-100, 22-23 October 1998.

46. Murphy, R.R., “Use of Scripts for Coordinating Perception and Action,” paper appears in Intelligent Robots and Systems 96, IROS 96, Proceedings of the 1996 IEEE/RSJ International Conference, v.1, pp. 156-161, 4-8 November 1996.

47. Ousterhout, J.K., “Scripting: Higher-Level Programming for the 21st Century,” *Computer,* v.31-3, pp. 23-30, March 1998.

48. Anupam, V. and Mayer, A., “Secure Web Scripting,” *IEEE Internet Computing,* v.2, issue 6, pp. 46-55, November-December 1998.

49. Sander, T. and Tschudin, C.F, “Towards Mobile Cryptography,” paper appears in Security and Privacy, Proceedings of the 1998 IEEE Symposium, pp. 215-224, 3-6 May 1998.

THIS PAGE INTENTIONALLY LEFT BLANK

INITIAL DISTRIBUTION LIST

No. Copies

1. Defense Technical Information Center.......... 2
   8725 John J. Kingman Road, Ste 0944
   Ft. Belvoir, VA 22060-6218

2. Dudley Knox Library.......... 2
   Naval Postgraduate School
   411 Dyer Rd.
   Monterey, CA 93943-5101

3. Chairman, Code CS.......... 1
   Naval Postgraduate School
   Monterey, CA 93943-5101

4. Prof. James Bret Michael.......... 1
   Computer Science Department, Code CS/Mj
   Naval Postgraduate School
   Monterey, CA 93943-5000

5. Prof. Bert Lundy.......... 1
   Computer Science Department, Code CS/Ln
   Naval Postgraduate School
   Monterey, CA 93943-5100

6. Deniz Kuvvetleri Komutanligi.......... 1
   Personel Daire Baskanligi
   Bakanliklar
   Ankara, TURKEY

7. Deniz Kuvvetleri Komutanligi 1
Kutuphanesi
Bakanliklar
Ankara, TURKEY

8. Deniz Harp Okulu 2
Kutuphanesi
Tuzla
Istanbul, TURKEY

9. Yazilim Gelistirme Grup Baskanligi 1
Deniz Harp Okulu Komutanligi
Tuzla
Istanbul, TURKEY

10. LT. JG. Refik TUFEKCIOGLU 2
Visnelik Mahallesi
Savas Caddesi Savas Apartmani No: 123 Daire: 10
Visnelik
Eskisehir, TURKEY